

**UNIVERSIDAD AUTONOMA DE MADRID
ESCUELA POLITECNICA SUPERIOR**



Doble grado en Ingeniería Informática y en Matemáticas

TRABAJO FIN DE GRADO

**REESTRUCTURACIÓN DE LOS SISTEMAS ANALÍTICOS
DE UNA OPERADORA AL PRESERVAR EL NÚMERO DE
ABONADO TRAS UN CAMBIO DE TECNOLOGÍA**

Autora: Deborah Sabrina Gabisson

Tutor: Aitor Santamaría Merino

Ponente: Ana González Marcos

Enero de 2017

REESTRUCTURACIÓN DE LOS SISTEMAS ANALÍTICOS DE UNA OPERADORA AL PRESERVAR EL NÚMERO DE ABONADO TRAS UN CAMBIO DE TECNOLOGÍA

AUTOR: Deborah Sabrina Gabisson

TUTOR: Aitor Santamaría Merino

PONENTE: Ana González Marcos

**Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Enero de 2017**

Resumen

Este Trabajo Fin de Grado se ha realizado en la empresa tecnológica Everis que ofrece servicios de consultoría IT a diversas empresas. En particular, se ha centrado en el trabajo para una operadora, en la cual originalmente, un cambio de ADSL a fibra no conllevaba la preservación del número de abonado, causando notables inconvenientes al usuario. La operadora se ha planteado adaptar todos sus sistemas para garantizar este requerimiento de forma transparente y sin costes adicionales al cliente.

Esto se ha conseguido modificando de forma considerable sus sistemas informáticos corporativos. Como en la gran mayoría de los sistemas informáticos distribuidos en aplicaciones independientes en los que se mueve una gran cantidad de datos, una operadora distingue sus sistemas en Operacionales y Analíticos[1]. Los sistemas Operacionales realizan la operación diaria, volcando los datos generados en los sistemas Analíticos, donde reside la inteligencia y análisis estratégico de su contexto (Inteligencia de negocio). Una modificación de envergadura como la expuesta impacta tanto en los sistemas Operacionales, como en los Analíticos. Este proyecto se ha centrado en su impacto en estos últimos, el área de Business Intelligence.

Este documento recoge el proceso llevado a cabo para la consecución de estos objetivos. Como este proyecto comprende todas las etapas del ciclo de vida de un proyecto de Ingeniería del Software, se explicará lo hecho en cada una de las distintas fases.

Mediante el uso del lenguaje de programación PL/SQL, se han desarrollado y adaptado los procesos ETL (del inglés *Extract, Transform and Load*). Así mismo se ha adaptado el Universo de Business Intelligence mediante el uso de tecnologías como el entorno de Business Objects para poder procesar la nueva funcionalidad de forma adecuada.

La finalidad era conseguir que la inteligencia estratégica y sus sistemas sean capaces de articular los cambios propuestos de forma coherente y sin comprometer la calidad del dato.

Tras la realización de este proyecto, se ha conseguido ese objetivo, y el proyecto está ya funcionando correctamente en los sistemas del cliente.

Palabras clave

Business Intelligence, ETL, Ingeniería del Software, Oracle, PL/SQL, Sistemas Analíticos.

Abstract

This Bachelor's Thesis was done in Everis, a multinational consulting firm offering IT solutions to other companies.

In particular, it is focused on the work for a phone company, in which originally, a migration from ADSL technology to Fiber didn't involve the preservation of the subscriber's number, causing significant inconveniences to the client. The company wanted to adapt its systems to ensure this requirement in a transparent way and without additional costs to the client.

This was achieved by modifying significantly their corporate computer systems. Like the vast majority of distributed computer systems in which a large amount of data is generated, this phone company distinguishes its systems in Operational and Analytical systems[1]. Operational systems perform daily operations, dumping the generated data in the analytical systems, where the intelligence and strategic analysis resides. A major modification as the exposed here impacts on both, operational and analytical systems. This project is focused on the second ones, in what is called Business Intelligence.

This document describes the process carried out to achieve these objectives. Since this project involves all stages of the life cycle of a software engineering project, the different phases of the life cycle will be explained.

Using the programming language PL/SQL, the ETL (Extract, Transform and Load) processes have been developed and adapted. The Business Intelligence Universe has been adapted as well to fulfill the client's needs, using Business Objects.

The aim was to get the analytical systems to be able to articulate the proposed changes consistently and without compromising data quality.

After the completion this project, the goal was achieved, and the project is properly working on the client's systems.

Keywords

Business Intelligence, ETL, Software Engineering, Oracle, PL/SQL, Analytic Systems.

Agradecimientos

Principalmente a mi madre, por su apoyo incondicional e infinita ayuda. Sin ella, nada de esto hubiera sido posible.

A mi tutor y a mi ponente por dedicarme tiempo y la oportunidad de realizar este TFG. Así como a aquellos profesores que me han ayudado durante la carrera.

Y a mis amigos, por regalarme tan buenos momentos durante estos años y hacerme sentir una persona afortunada.

INDICE DE CONTENIDOS

| | | |
|-------|--|----|
| 1 | Introducción..... | 1 |
| 1.1 | Motivación..... | 1 |
| 1.2 | Objetivos..... | 2 |
| 1.3 | Organización de la memoria..... | 3 |
| 2 | Estado del arte | 5 |
| 2.1 | Sistemas impactados..... | 5 |
| 2.2 | Lógica de un sistema | 6 |
| 2.3 | Módulos..... | 7 |
| 3 | Metodología y herramientas | 9 |
| 3.1 | Metodología..... | 9 |
| 3.1.1 | Fases del proyecto | 10 |
| 3.1.2 | Planificación | 10 |
| 3.2 | Herramientas..... | 11 |
| 3.2.1 | Entorno | 11 |
| 3.2.2 | Almacenamiento y control de versiones..... | 12 |
| 4 | Análisis | 13 |
| 4.1 | Requisitos | 13 |
| 4.2 | Impactos | 13 |
| 4.2.1 | Tablas impactadas..... | 13 |
| 4.2.2 | Cadenas impactadas..... | 14 |
| 4.3 | Análisis de los requisitos | 14 |
| 5 | Diseño..... | 17 |
| 5.1 | Diseño funcional..... | 17 |
| 5.1.1 | Cobertura de requisitos y solución | 17 |
| 5.1.2 | Listado de sub-componentes | 17 |
| 5.1.3 | Descripción de los procesos | 18 |
| 5.1.4 | Carga inicial y normalización de datos..... | 21 |
| 5.2 | Flujogramas | 21 |
| 5.3 | Acuerdo de interfaz | 23 |
| 5.4 | Planificación de ejecución..... | 24 |
| 6 | Desarrollo | 25 |
| 6.1 | Instalación, carga inicial y normalización | 25 |
| 6.2 | Desarrollo de componentes | 26 |
| 6.2.1 | Cadena de fibra..... | 26 |
| 6.2.2 | Cadena de cobre..... | 28 |
| 6.3 | Marcha Atrás | 29 |
| 6.4 | Universo Business Objects | 30 |
| 7 | Integración, pruebas y resultados | 31 |
| 7.1 | Pruebas de despliegue..... | 31 |
| 7.2 | Pruebas de marcha-atrás | 33 |
| 7.3 | Pruebas progresivas | 33 |
| 7.4 | Pruebas regresivas | 34 |
| 7.5 | Pruebas de rendimiento | 35 |
| 8 | Conclusiones y trabajo futuro..... | 37 |
| 8.1 | Conclusiones..... | 37 |
| 8.2 | Trabajo futuro | 38 |
| | Referencias | 39 |
| | Glosario | 41 |

| | |
|--------------------------------------|-----|
| Acrónimos | 43 |
| Anexos | XLV |
| A Manual de despliegue | XLV |
| Instalación y configuración | II |
| Borrado de tablas backups | II |
| Descripción de la marcha atrás | III |

INDICE DE FIGURAS

| | |
|---|----|
| FIGURA 1-1: FUNCIONAMIENTO DE LOS SISTEMAS. | 1 |
| FIGURA 2-1: ESTRUCTURA DEL DATA WAREHOUSE | 5 |
| FIGURA 2-2: EJECUCIÓN DE UNA CAJA. | 7 |
| FIGURA 2-3: ESTRUCTURA DE UNA CADENA | 7 |
| FIGURA 2-4: MÓDULOS | 8 |
| FIGURA 3-1: METODOLOGÍA EN V | 9 |
| FIGURA 3-2 PLANIFICACIÓN | 11 |
| FIGURA 4-1: SISTEMAS IMPACTADOS | 15 |
| FIGURA 5-1: FLUJOGRAMA REQUISITO 1 | 22 |
| FIGURA 5-2: FLUJOGRAMA REQUISITO 2 | 22 |
| FIGURA 5-3: FLUJOGRAMA REQUISITO 3 CADENA DE FIBRA. | 23 |
| FIGURA 5-4: FLUJOGRAMA REQUISITO 3 CADENA DE COBRE | 23 |
| FIGURA 5-5: EJECUCIÓN CAJAS DE LA CADENA DE FIBRA | 24 |
| FIGURA 7-1: PRUEBAS DE DESPLIEGUE, CREACIÓN TABLA TEMPORAL | 32 |
| FIGURA 7-2: PRUEBAS DE DESPLIEGUE, DATOS TABLA TEMPORAL | 32 |
| FIGURA 7-3: PRUEBAS DE DESPLIEGUE, CREACIÓN DE CAMPOS. | 33 |
| FIGURA 7-4: PRUEBAS PROGRESIVAS, SITUACIÓN ACTUAL CARTERA. | 34 |
| FIGURA 7-5: PRUEBAS PROGRESIVAS, SITUACIÓN ACTUAL HISTÓRICO. | 34 |
| FIGURA 7-6: PRUEBAS PROGRESIVAS, RESULTADO EN CARTERA. | 34 |
| FIGURA 7-7: PRUEBAS PROGRESIVAS, RESULTADO EN EL HISTÓRICO | 34 |

INDICE DE TABLAS

| | |
|--|-----|
| TABLA 5-1: COBERTURA DE REQUISITOS. | 17 |
| TABLA 5-2: LISTADO DE SUBCOMPONENTES..... | 18 |
| TABLA 5-3: ACUERDO DE INTERFAZ PARA LA CARTERA. | 24 |
| TABLA 5-4: ACUERDO DE INTERFAZ PARA EL HISTÓRICO DE SOLICITUDES..... | 24 |
| TABLA 0-1: ARCHIVOS DE INSTALACIÓN Y EJECUCIÓN..... | XLV |
| TABLA 0-2: ARCHIVOS DE CONFIGURACIÓN. | XLV |
| TABLA 0-3: CÓDIGO FUENTE..... | XLV |
| TABLA 0-4: ARCHIVOS MARCHA ATRÁS..... | II |

1 Introducción

1.1 Motivación

Hoy en día la economía se caracteriza por el rápido y continuo cambio de los mercados y de las oportunidades de negocio. Por ello, para tener éxito, es importante que la empresa tome las decisiones de negocio correctas, y de forma rápida. Las decisiones de negocio se toman partiendo de una base de análisis de la situación actual y pasada de la empresa.

Estos datos que se analizan son proporcionados por los sistemas operacionales, los cuales realizan la operación diaria. Pero los datos suelen ser heterogéneos, pues suelen provenir de diferentes distribuidores (en nuestro caso, de diferentes provisionadores¹), e implementados en diferentes tecnologías, y suele haber autonomía de los datos, lo que implica que normalmente éstos sean independientes unos de los otros. [2]

Los sistemas de data warehouse (traducido como almacén de datos) se han convertido en un componente esencial de los sistemas informáticos corporativos de las empresas. Éstos juegan un papel crucial en los sistemas analíticos, en los que reside la inteligencia y análisis estratégico (inteligencia de negocio). Mediante la recolección y la consolidación de los datos provenientes los sistemas operacionales, los data warehouse proporcionan información homogénea como base para la planificación y toma de decisiones de una empresa. [3]

El proceso mediante el cual los sistemas analíticos se alimentan de la información procedente de los sistemas operacionales existentes, se llama proceso de Extracción, Transformación y Carga de datos (ETL). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, usualmente en el data warehouse. [4]

Los sistemas analíticos permiten analizar datos históricos, optimizar el rendimiento empleando un entorno diferente del operacional e integrar datos consolidados de diferentes aplicaciones operacionales. [5]

El esquema del funcionamiento de los sistemas analíticos se muestra en la Figura 1-1.

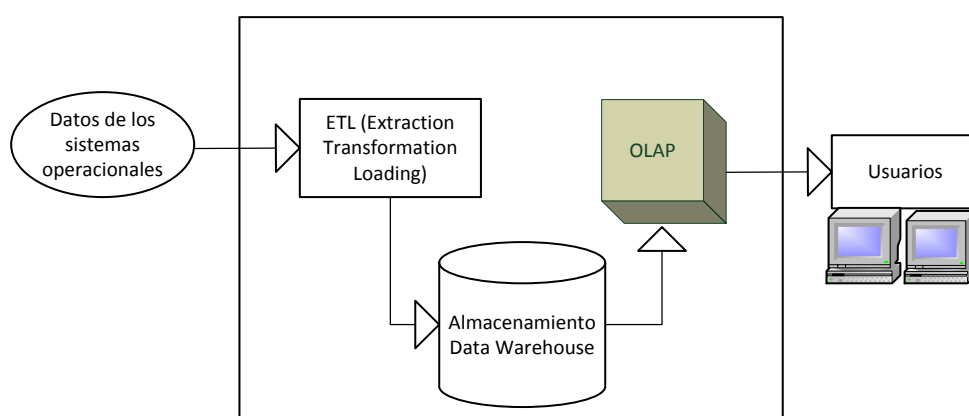


Figura 1-1: Funcionamiento de los sistemas analíticos.

¹ Sistema que se encarga de hacer la provisión de un servicio o producto.

Entre la base de datos de la empresa y el usuario final, que será el que explote los datos del data warehouse (será quien realizará los informes), debe haber una capa para que éste pueda acceder a los datos sin necesidad de conocer las complejidades de la base de datos, o dónde se encuentra ésta. En nuestro sistema, por ser los datos explotados con el programa Business Objects, esta capa semántica será un Universo de Business Objects (abreviado BO). Un universo de BO es una representación de los datos del data warehouse creado usando términos de negocio simples para describir el entorno de negocio y permitir al usuario que extraiga exactamente la información que le interesa. [6][7]

Con los avances continuos de la tecnología, los sistemas informáticos corporativos se deben adaptar a estos cambios. Además de adaptar los sistemas operacionales, se deben adaptar también los sistemas analíticos para evitar obtener datos erróneos e informes con información incorrecta por no ser ésta bien tratada.

Un ejemplo de un avance tecnológico es la implantación de un nuevo medio de transmisión, la fibra óptica, empleado habitualmente en redes de datos y telecomunicaciones. Los sistemas analíticos de la operadora en cuestión (una de las más importantes del mercado actual), no están diseñados para soportar dicho tipo de avances. Soportan todo tipo de oferta con ADSL o con fibra, pero no soportan un cambio de una a otra. En este proyecto se adaptarán los sistemas analíticos para soportar dicho cambio. Nos concentraremos en el área de Business Intelligence, adaptando los procesos ETL y los universos Business Objects.

1.2 Objetivos

Actualmente, para la provisión de internet para líneas fijas, la operadora en cuestión ofrece dos tecnologías: ADSL y fibra óptica, procediendo cada una de un provisionador diferente. Por ello y por cómo están diseñados los sistemas actuales, no es posible que el cambio de un cliente que esté en tecnología ADSL a tecnología fibra, se refleje de forma adecuada en los sistemas analíticos. La única posibilidad para poder mover a un cliente de ADSL a fibra es realizar un alta de servicio en fibra y una baja del servicio antiguo que tiene en ADSL, provocando una baja y un alta del contrato.

Esta solución no es una solución óptima. El cliente pierde el número de teléfono, pues no se pueden realizar acciones coordinadas para el alta en fibra y la baja en ADSL (por proceder de diferentes provisionadores), así como las ventajas por antigüedad para el cliente. Así mismo, para la compañía supone un análisis erróneo de los datos debido a que el número de bajas, de altas y de cambios de tecnologías no sería el número real.

Se solicita adaptar los sistemas informáticos corporativos para garantizar que el cambio de tecnología de ofertas ADSL a fibra no suponga un alta y una baja del contrato, sino que haya una unificación en un solo contrato en los cambios de provisionador. Es decir, que se refleje como un cambio de tecnología, y que se realice de forma transparente y sin impactos en costes ni experiencia del cliente.

Al considerarse ahora como un cambio de tecnología, también se solicita un indicador de migración² (indica si una solicitud es una migración o no) y la fecha de la última migración.

Así mismo, el cliente ha solicitado implementar una solución para la trazabilidad de las solicitudes por las que ha pasado un cliente. Y, además de ello, se ha solicitado incluir el motivo de cancelación del contrato en el universo de Business Objects.

² Cuando se hable de migración se referirá a cambio de tecnología de cobre (ADSL) a fibra óptica.

Esta modificación en los sistemas analíticos provocará una mejora en la retención de clientes que cambian de tecnología de ADSL a fibra. Así mismo, una vez modificados tanto los sistemas operacionales³ como los analíticos, se le podrá ofrecer al cliente un mejor servicio sin pérdida de ventajas por antigüedad, ni pérdida del número de teléfono, resultando esto en una mayor satisfacción del cliente.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1. Introducción: motivación y objetivos del proyecto.
- Capítulo 2. Estado del arte. En este capítulo se describe la estructura lógica de un sistema y los sistemas impactados por los cambios.
- Capítulo 3. Metodología y herramientas. Se describe la metodología seguida durante el ciclo de vida del proyecto y las herramientas que se han utilizado para llevarlo a cabo.
- Capítulo 4. Análisis. Se realiza un estudio de los sistemas impactados, así como de los requisitos del proyecto.
- Capítulo 5. Diseño. En este capítulo se describe el diseño de la solución a implementar.
- Capítulo 6. Implementación. Se explica cómo se han implementado los distintos módulos.
- Capítulo 7. Pruebas. Se detallan las pruebas llevadas a cabo sobre el proyecto que permiten validar y verificar el correcto funcionamiento del mismo.
- Capítulo 8. Conclusiones y trabajo futuro. Se explican las conclusiones a las que se ha llegado, así como posibles mejoras del sistema.

³ Los cambios en los sistemas operacionales son realizados por otra empresa. En esta empresa sólo llevamos los sistemas analíticos y el área de Business Intelligence.

2 Estado del arte

2.1 Sistemas impactados

Los sistemas de la operadora en cuestión son abundantes, pero para este proyecto sólo son relevantes dos, los dos que están relacionados con la telefonía fija (todos los demás están ligados a móvil, y no influyen en este cambio): el Data Warehouse y el Analítico del Fijo.

El Data Warehouse del negocio fijo contiene información sobre clientes, productos, servicios, contratos, histórico de solicitudes, provisionadores etc. Tiene interfaz con el provisionador de cobre y de fibra⁴. El provisionador de cobre es el sistema que informará las solicitudes de tecnología cobre (ADSL). El provisionador de fibra es el sistema que informará las solicitudes de tecnología fibra. Con los datos que le vienen de éstos, genera dos entidades principales, la cartera de clientes y el histórico de solicitudes. Es el sistema que tiene la lógica. Está implementado con tecnología de Base de Datos Oracle (y PL/SQL) [8] y los datos se explotan con Business Objects [9]. La estructura del Data Warehouse se muestra en la Figura 2-1.

El Analítico del Fijo es una copia del Data Warehouse. No tiene lógica, ni tiene interfaz directa con los provisionadores. Sino que toda la información proviene del Data Warehouse. Inicialmente, existía solo el Data Warehouse, pero se decidió que se quería tener una base de datos con tecnología Teradata [10] y explotar los datos con SAS [11] para obtener mayor rendimiento y velocidad [12]. Por ello se construyó el sistema Analítico del Fijo.

El cambio a realizar impacta en estos dos sistemas. Este Proyecto de Fin de Carrera se ha centrado en el sistema Data Warehouse, mientras que otro equipo de trabajo con sede en Murcia se ha centrado en el Analítico del Fijo.

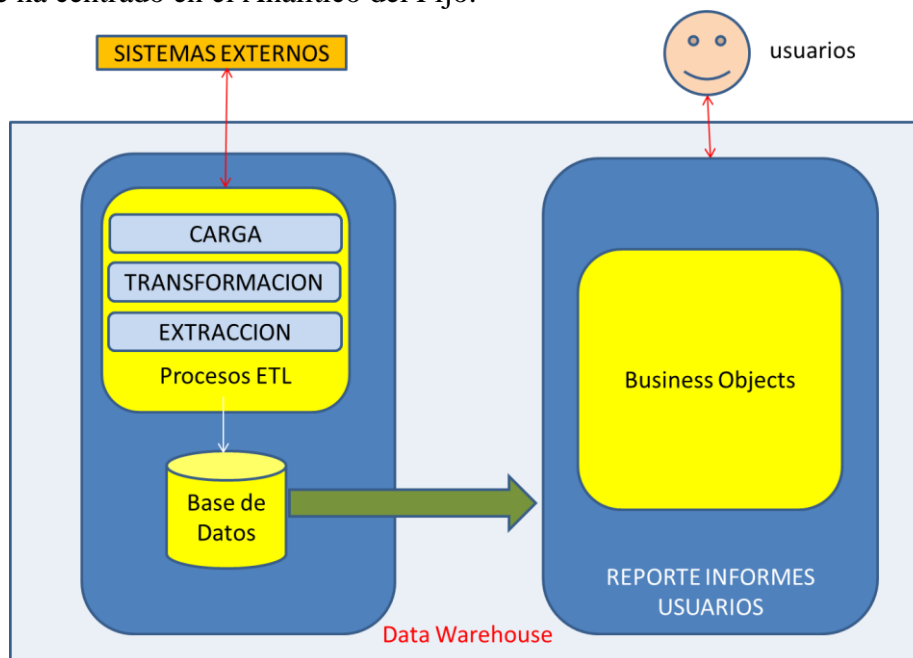


Figura 2-1: Estructura del data warehouse

⁴ Cuando se hable de tecnología cobre se referirá a ADSL. Y cuando se hable de fibra, a fibra óptica.

2.2 Lógica de un sistema

Los sistemas analíticos de esta operadora alimentan sus datos mediante cadenas de procesos *batch*, procesos ejecutados de forma planificada y sin intervención del usuario. En un sistema el cual implemente varias funcionalidades simultáneamente y cada una de estas funcionalidades posea una serie de procesos ETL, se tendrá una cadena de procesos *batch* por cada funcionalidad (lo abreviaremos cadena).

Cada cadena se compone de una secuencia de sus respectivas cajas. Una caja es un *link* de Unix que hace referencia a un proceso concreto, por ejemplo, proceso1.caja. Estas cajas mantienen un orden de ejecución, generando dependencias entre ellas. Una caja es un *link* a su script correspondiente, el fichero proceso1.sh.

Este script proceso1.sh ejecuta el fichero proceso1.sql correspondiente, que es el cual contiene el código y la lógica del proceso a realizar.

Así mismo, se tienen ficheros de configuración de los procesos, que son ficheros que poseen extensión .cfg y tienen las variables de entorno que podrán ser utilizadas por los procesos de carga. Dentro de los ficheros de configuración tenemos:

- Fichero de metadatos (metadatos_tabla.cfg): se desarrolla de acuerdo al diseño de metadatos. Existe una serie de tablas de metadatos que almacenan constantes y variables que son utilizadas como variables de entorno por las distintas cajas (procesos).
- Fichero común a la cadena (nombre_cadena.cfg): define las variables de entorno para indicar la localización de los ficheros relacionados con la cadena.
- Fichero propio de la caja (nombre_caja.cfg): define las variables propias de esta caja de la cadena y hereda las rutas definidas en el fichero de configuración de la cadena.

Tras la ejecución de una cadena, se generan ficheros *log* con la ejecución detallada de los procesos (extensión .log).

El flujo de la ejecución de una caja se puede ver representada en la Figura 2-2, y la estructura de una cadena en la Figura 2-3.

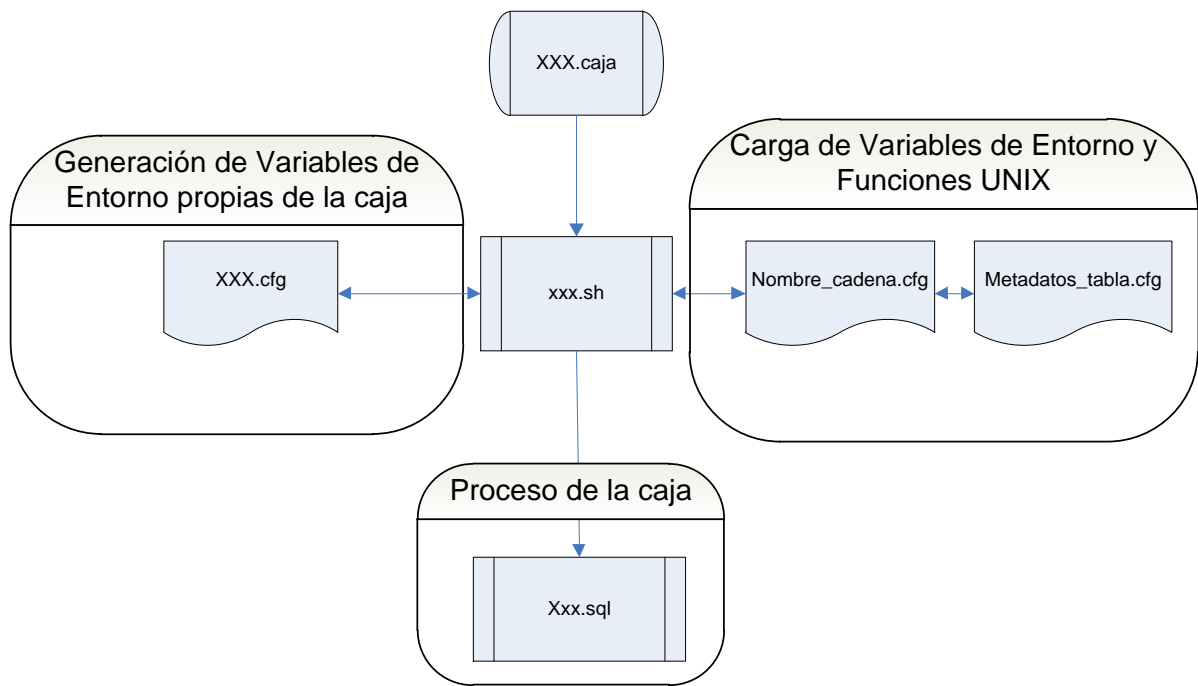


Figura 2-2: Ejecución de una caja.

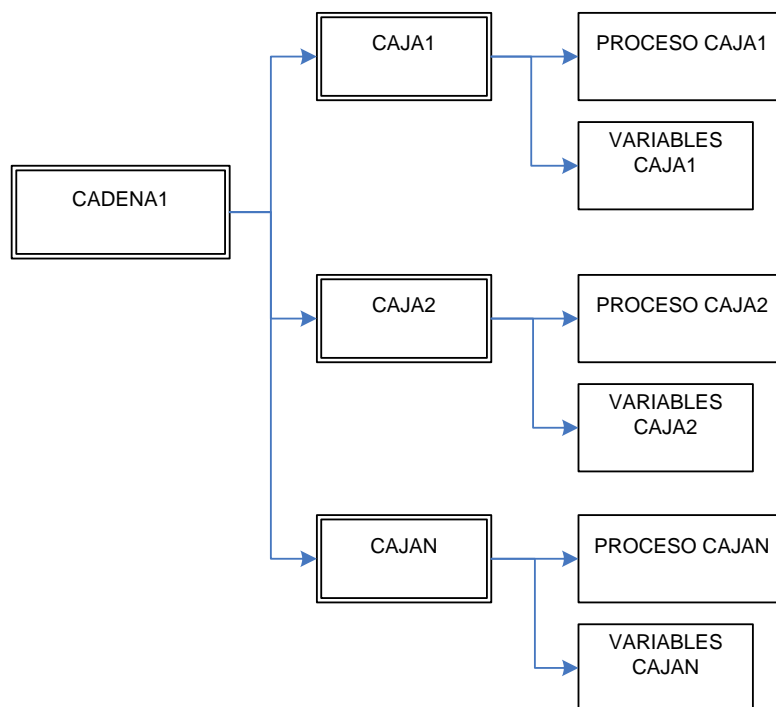


Figura 2-3: Estructura de una cadena

2.3 Módulos

Existen tres módulos claramente diferenciados, y estos módulos interaccionan entre sí mediante llamadas a funciones y procedimientos generando el flujo de información deseado. Estos módulos se pueden observar en la Figura 2-4.

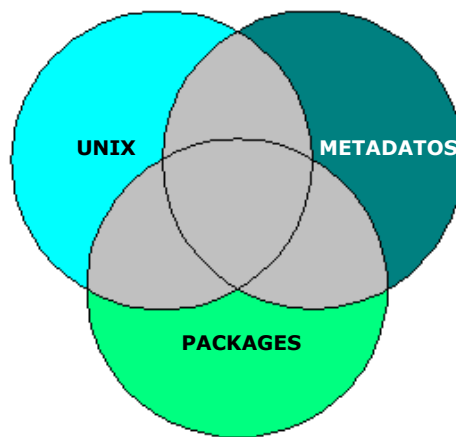


Figura 2-4: Módulos

Módulo metadatos

Estas variables mantendrán su valor en las diversas llamadas a los procedimientos o funciones y servirán para evitar realizar el envío de datos por medio de parámetros en las llamadas a distintos scripts o utilización de variables globales almacenadas en el sistema operativo.

Módulo *packages*

La finalidad de este paquete de base de datos es permitir el desarrollo de procesos sin necesidad de repetir código desarrollado en otros procesos anteriormente, es decir, los procesos que son comunes para varias cajas o cadenas del sistema deberán ir almacenados en un paquete.

En este proyecto, no se ha hecho uso de ningún proceso de los *packages*.

Módulo UNIX

El módulo UNIX es el encargado de iniciar cualquier tipo de proceso ETL.

3 Metodología y herramientas

3.1 Metodología

Debido al tiempo disponible para el desarrollo de este proyecto, las entregas parciales de documentación al cliente, requisitos iniciales correctos y completos, y habiendo validación al final de cada etapa, se ha decidido seguir la metodología en V. [13]

La retroalimentación ha permitido volver hacia atrás en caso de necesitar realizar modificaciones tras una validación o verificación. Esta metodología se representa en la Figura 3-1.

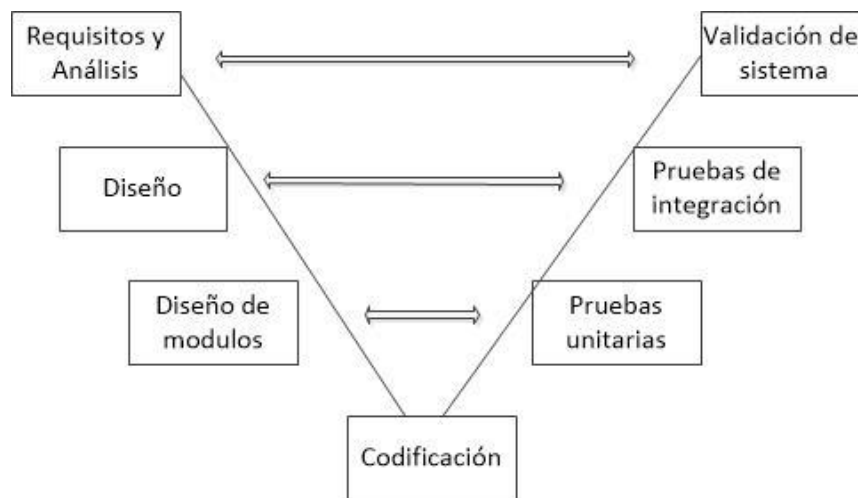


Figura 3-1: Metodología en V.

Para la realización del proyecto, se disponía de tres entornos: un entorno de desarrollo, uno de test o integración, y el entorno de producción.

La fase de implementación y pruebas se ha realizado de la siguiente manera: se implementaba un módulo (una funcionalidad), se subía al entorno de desarrollo, y se realizaban las pruebas unitarias de dicho módulo, ejecutando individualmente la caja correspondiente a dicha funcionalidad.

Una vez realizada la implementación y las pruebas unitarias de todos los módulos, se llevaba a cabo otra prueba ejecutando la cadena entera en el entorno de desarrollo. Si el resultado era satisfactorio, se subía el código al entorno de integración y se hacían las pruebas integradas y regresivas.

El entorno de producción es el entorno en el cual se encuentra el sistema en funcionamiento. Para ese entorno se han utilizado usuarios de lectura durante todo el proyecto. Sólo al finalizarlo, al poner el proyecto en funcionamiento, se han usado usuarios con permisos de escritura. Esto último lo ha realizado el equipo de mantenimiento.

A partir de las fuentes de la línea base de producción se ha realizado el desarrollo. Así mismo, es dónde se ha investigado el funcionamiento de los distintos sistemas, cadenas y cajas, pues además de contener todo el sistema en sí, contiene los ficheros *log* con información sobre la ejecución de los procesos, debido a que no existe una documentación fiable sobre el diseño de los módulos objeto.

Se han realizado reuniones de coordinación periódicas con el centro de Murcia para regular el acceso a los entornos, los datos insertados para las pruebas, el código, etc. En el apartado 8. Conclusiones y trabajo futuro, se redactan algunos problemas afrontados relacionados a este asunto.

3.1.1 Fases del proyecto

Durante la fase de análisis se ha realizado el estudio de la estructura de los sistemas analíticos de la compañía en cuestión. Así mismo, se ha realizado un estudio de los requisitos del proyecto solicitados por el cliente. Una vez se han comprendido éstos, se ha realizado el análisis de los sistemas impactados por el proyecto, y se ha redactado el documento con la solución a alto nivel para entregarle al cliente, el cual debía aprobar antes de poder empezar con la fase de diseño.

En la fase de diseño, se ha realizado el diseño funcional del proyecto. Se ha realizado el acuerdo de interfaz entre los dos sistemas impactados, los flujogramas de los diferentes procesos afectados por el proyecto, y un documento con el diseño funcional detallado del proyecto, el cual también ha sido aprobado por el cliente antes de pasar a la fase de implementación.

Durante la fase de implementación se ha realizado el desarrollo de los distintos componentes. Al finalizar la implementación, y tras el resultado positivo de las pruebas, se han realizado dos archivos con extensión .tar que contienen el resultado de los archivos fuente modificados: uno para poder realizar el pase a producción integrando nuestro proyecto en los sistemas actuales, y otro de marcha atrás, el cual se ejecutaría en caso de que algo fallase dejando el sistema en su estado anterior. Así mismo, se ha documentado como se debe realizar el pase y como se debe realizar la marcha atrás en caso de necesidad en el documento “Notas para el pase a producción”, para el equipo de mantenimiento, encargado de realizar el despliegue en producción.

En la fase de pruebas se han realizado las pruebas de instalación, las de marcha atrás, las unitarias, las de integración, y las pruebas regresivas. Evidenciando esto, se han realizado dos documentos, el *Test Plan* y el *Test Report* [14]. Así mismo se ha realizado una estimación de base de datos. Toda esta información es requerida por el equipo de mantenimiento para poder realizar el despliegue.

3.1.2 Planificación

La planificación se ha realizado de acuerdo a las fechas de entrega, tanto de la documentación, como de la implantación del proyecto en el sistema del cliente.

Lo primero fue entregar el documento solución, el cual se realizó después de un breve análisis de los requisitos y del sistema, presentando en él la solución a lo solicitado por el cliente a alto nivel, así como las limitaciones de la solución, el alcance del proyecto y un calendario del mismo, entre otras cosas.

Tras la aprobación de éste, se realizó el análisis y el diseño, siendo estas dos fases del ciclo de vida las que más tiempo han consumido debido a la complejidad del sistema y el tamaño del mismo. Como resultado de estas fases, se realizó la entrega del documento del diseño funcional de la solución, el acuerdo de interfaz con otros sistemas, así como el plan de pruebas.

Una vez realizado esto, comenzó la fase de implementación y pruebas, realizando la implementación de los distintos módulos y las pruebas unitarias de éstos, para luego poder

realizar las pruebas integradas tanto progresivas como regresivas. Se generaron los instalables de marcha adelante y marcha atrás, la estimación de bases de datos, el reporte de pruebas, y la documentación para el equipo de servicio o mantenimiento.

Debido a que el código debe ser aprobado por un encargado de control de calidad del cliente, éste se ha debido modificar repetidas veces, de forma que los procesos tuvieran el mayor rendimiento posible, y dicho responsable lo aprobara. Lo mismo ocurrió con la estimación de bases de datos, en la cual se debía rentabilizar el espacio.

Una vez implantado el proyecto, se realizó la memoria del TFG.

Todas las fases anteriormente comentadas se reflejan en el diagrama de la Figura 3-2.

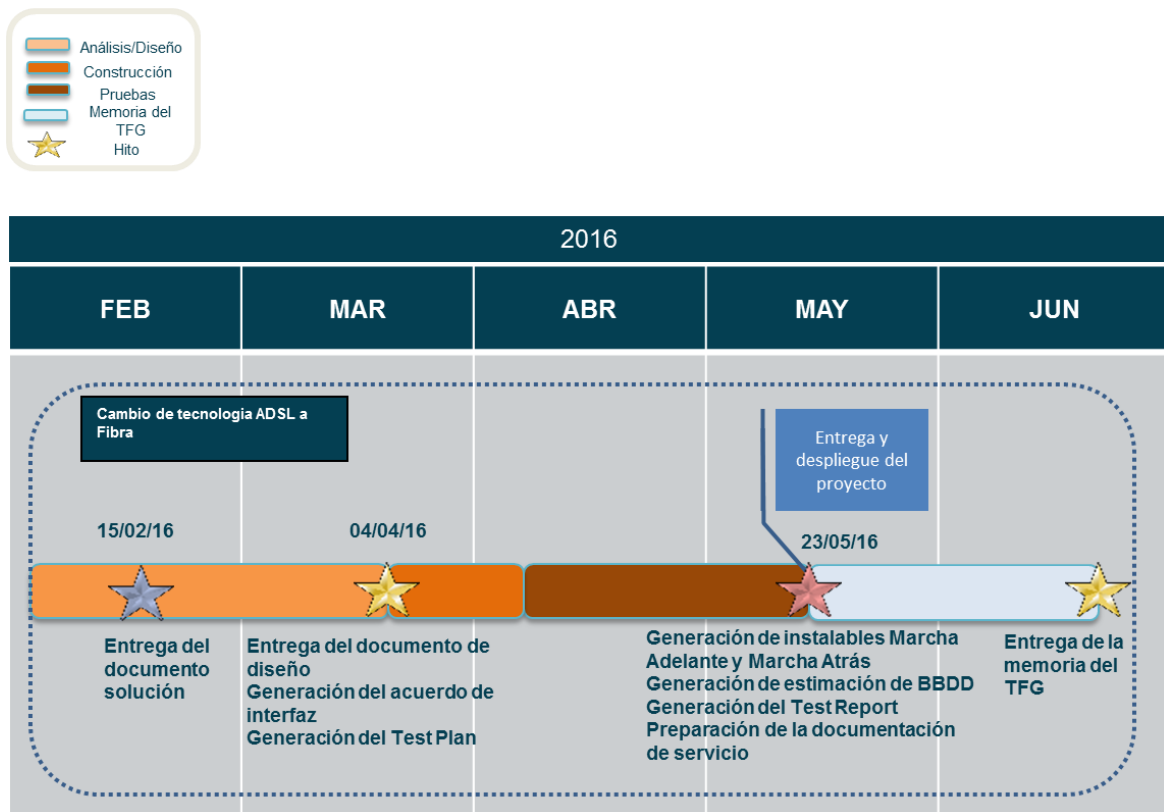


Figura 3-2: Planificación.

3.2 Herramientas

3.2.1 Entorno

Para la realización de este proyecto se ha hecho uso de las siguientes herramientas:

- Cliente de base de datos Oracle [15].
- Para la conexión a los servidores remotos del cliente se ha utilizado VPN Client [16] y Cisco Network Access [17].
- Para el desarrollo de código, se han utilizado los entornos de desarrollo PL/SQL Developer [18].

- Para la modificación del universo, se ha utilizado Business Objects, la herramienta con la cual se explotan los universos del sistema Data Warehouse. En particular, se ha modificado el universo con Business Objects Developer.
- Para la conexión a las máquinas de Unix en remoto de los diferentes entornos, se ha hecho uso de la herramienta Putty, un cliente Telnet y SSH remoto [19].
- Para subir o descargar archivos de las máquinas remotas se ha utilizado un cliente FTP, Filezilla [20].

Lo primero que se ha hecho ha sido instalar el cliente de Oracle. Una vez instalado el cliente, se ha creado un fichero *hosts*, el cual contiene las asignaciones de las direcciones IP a los nombres de los *host* (la dirección IP del servidor con el cual queremos conectar), y se ha creado un archivo TNSNAMES.ORA [21] que es un archivo de configuración que contiene descriptores de conexión que describen los servicios de destino y la información de ruta de red que el conector necesita para establecer conexiones con las bases de datos Oracle.

Para la conexión a las máquinas remotas, ya sea para acceder a la máquina de Unix o a las bases de datos desde PL/SQL Developer, primero es necesario realizar la autenticación de los usuarios para el acceso a la red. Esto se realiza a través del *Network User Authentication* de Cisco. Seguidamente, es necesario acceder a la red privada virtual, VPN (del inglés Virtual Private Network). Esto se ha realizado a través de la herramienta VPN Client.

Una vez realizado esto, ya se puede acceder a través de la herramienta Putty a los distintos servidores remotos del cliente. Para acceder a las bases de datos, se puede acceder desde la misma máquina de Unix, o se puede usar la herramienta PL/SQL Developer. Dependiendo de la finalidad, se usarán las herramientas de desarrollo o la terminal de Unix a través de SQL Plus [22], una herramienta de Oracle para interactuar con la base de datos mediante línea de comandos.

3.2.2 Almacenamiento y control de versiones

Como método de almacenamiento y de control de versiones se ha decidido utilizar *Apache Subversion* (abreviado SVN) [23], corporativo de Everis, una herramienta de control de versiones *open source*. Como cliente *subversion*, se ha utilizado TortoiseSVN [24]. El repositorio se ha almacenado en los servidores de Everis. De esta forma, se ha conseguido almacenar y sincronizar archivos en línea y compartirlos con el equipo de Murcia.

La gestión de recursos se ha realizado a través de la herramienta Jira [25]. Con Jira se ha realizado la distribución de las tareas en la línea de tiempo del proyecto, se ha asignado cada una de las tareas a miembros del equipo para el seguimiento de las mismas, y se ha realizado la gestión del estado de avance de las tareas mediante la imputación de tiempo empleado en cada una de las acciones ejecutadas sobre la tarea.

4 Análisis

Durante esta fase se ha realizado un estudio de los requisitos del proyecto solicitados por el cliente. Una vez se han comprendido éstos, se ha realizado el estudio de la estructura de los sistemas, deduciendo así los sistemas impactados por el proyecto. Tras repartir el trabajo por sistemas con el equipo de trabajo de Murcia, se han analizado las cadenas y cajas impactadas por este proyecto. Para ello, además de la documentación disponible, se ha leído el código de los diferentes procesos *batch* y se han realizado pruebas de concepto, ayudando así a realizar un mejor análisis y obtener un mayor conocimiento del funcionamiento del sistema.

Aquí se resumirán brevemente los requisitos más importantes, unificados para mayor claridad, y se explicarán los impactos en los sistemas.

4.1 Requisitos

R1. Unificación en un solo contrato los cambios de proveedor: se necesita que los cambios de proveedor de cobre a fibra no supongan en los analíticos una baja y un alta del contrato.

R2. Identificación en el histórico de solicitudes el cambio de tecnología de cobre a fibra, además de la fecha de la última migración.

R3. Implementar una solución para la trazabilidad de las solicitudes por las que ha pasado un cliente.

R4. Incluir en el universo de BO un campo para el motivo de la cancelación del contrato.

El desarrollo de las modificaciones necesarias para cumplir los requerimientos puede producir un aumento en los tiempos de ejecución de ambas cadenas. Aunque no sea un requisito expreso, este aumento debe ser el menor posible, sin penalizar demasiado la ejecución de las demás cadenas.

4.2 Impactos

Como se ha comentado anteriormente, se ven impactados dos sistemas, el Data Warehouse del Fijo, y el Analítico del Fijo. El alcance del proyecto son los cambios en el Data Warehouse del Fijo.

Tras el análisis, se ha visto que para cumplir con los requisitos solicitados por el cliente de la manera más óptima se deben modificar las dos tablas principales: la tabla de la cartera de clientes, llamada Cartera, y la tabla del histórico de solicitudes, llamada Historico.

4.2.1 Tablas impactadas

- **Tabla Cartera:** esta tabla contendrá la información de los clientes que se encuentran en la cartera de clientes, esto es, tanto los clientes que se han dado de alta, como los que han sido dados de baja, junto con la información de los productos que tienen contratados actualmente (Oferta Actual). Debería haber una entrada por cada cliente.
- **Tabla Historico:** tabla del histórico de ofertas. Se mantiene información de cada solicitud que se ha tramitado. Es decir, contiene cada oferta por la que haya pasado un cliente.

En ambas tablas, cada oferta lleva asociada una solicitud y cada solicitud va relacionada con una solicitud padre (campo `id_solicitud_padre`) que indica la solicitud anterior a ésta que ha realizado el cliente. De esta manera, se puede saber cronológicamente cuáles han sido las solicitudes por las que ha pasado un cliente.

Así mismo, ambas tablas tienen el campo `tipo_solicitud` que indica el tipo de solicitud que es: alta, baja, cambio de titular, cambio de oferta, etc. Un cambio de tecnología de cobre a fibra se identifica actualmente con el `id_solicitud_padre = -1`, y el tipo de solicitud a 20.

4.2.2 Cadenas impactadas

Como ya se ha mencionado anteriormente, entre los sistemas fuentes para el Data Warehouse del Fijo, está el provisionador de cobre y el provisionador de fibra. De las 120 cadenas que hay en el Data Warehouse, tras el análisis, se llegó a la conclusión que hay impacto en dos, que son las cadenas diarias relativas a cada provisionador.

- **Cadena fibra:** la cadena del provisionador de fibra cargará la información de los nuevos clientes o la modificación de los ya existentes, con una frecuencia diaria y de forma incremental. Los datos de los clientes se obtienen de los sistemas operacionales del provisionador de fibra.
- **Cadena cobre (ADSL):** igual que la anterior, pero proveniente del provisionador de cobre (ADSL).

El hecho de modificar las dos tablas principales de este sistema, implica que se deberán modificar así mismo las dos cadenas principales, `cadena_fibra`, y `cadena_cobre`, que son las que realizan los procesos ETL cuya funcionalidad se quiere modificar.

La `cadena_fibra` cuenta con 65 cajas, y la `cadena_cobre` con 96. Se han analizado las diferentes cajas para así poder diseñar la solución mencionada en el apartado de Diseño.

4.3 Análisis de los requisitos

El primer requisito supone que en la tabla `Cartera` no se deben generar dos contratos, uno activo y otro de baja, este movimiento se debe tratar como un cambio de oferta en donde la `Cartera`, una vez cerrado el proceso de migración, solo mantendrá un contrato activo.

Para el segundo requisito, se creará en la tabla `Cartera` un campo indicador de migración, el cual indicará si se ha realizado una migración de cobre a fibra, y otro de fecha de última migración.

Para el tercero, se añadirá un identificador de la primera solicitud de alta generada al cliente, el campo `solicitud origen`, tanto a la tabla de `Cartera` y como a la del histórico de solicitudes.

Para el cuarto requisito, se incluirá en el universo de Business Objects del Data Warehouse un campo que indicará el motivo de cancelación del contrato, para el cual se realizará un mapeo de un campo que proviene directamente del provisionador de fibra.

En todo el proyecto, cuando se quiera identificar un cliente como unívoco (o los que interesa distinguir como entradas diferentes en la `Cartera`), se tomará como referencia los siguientes campos: el número de teléfono, el número de documento, el tipo de documento y el identificador del contrato. En realidad, bastaría con tres de éstos, pues el tipo de

documento no haría falta, pero como ya hay un índice creado con estos cuatro campos en las dos tablas principales, la Cartera y el histórico de solicitudes, se ha decidido incluirlo para optimizar las consultas, y no tener que crear un índice nuevo.

Un esquema de los sistemas impactados con los campos a añadir para la correcta implementación de los requisitos se puede ver en la Figura 4-1.

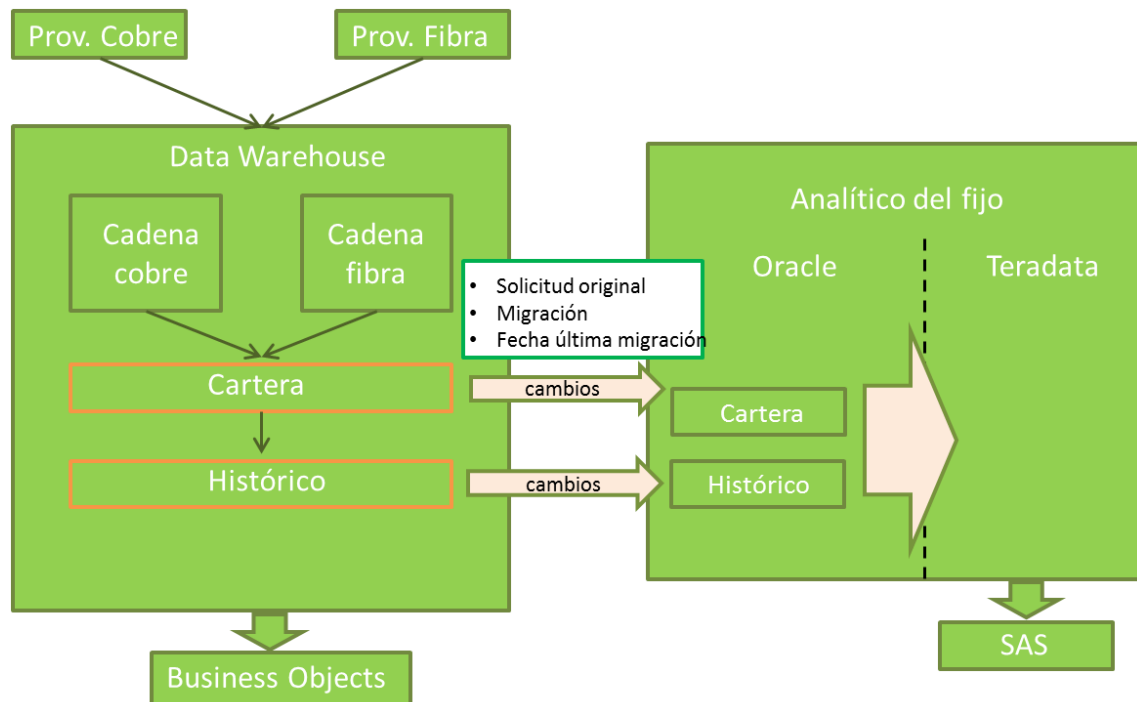


Figura 4-1: Sistemas impactados

5 Diseño

En esta fase se ha realizado el diseño funcional del proyecto, el cual se ha visto plasmado en un documento que se ha tenido que entregar al cliente antes de poder continuar a la siguiente fase. Así mismo, se ha realizado el contrato de interfaz entre los dos sistemas impactados y los flujogramas de los diferentes procesos afectados por el proyecto.

5.1 Diseño funcional

5.1.1 Cobertura de requisitos y solución

En la Tabla 5-1 se describe la cobertura de los requisitos según la solución adoptada.

| ID-RC | Nombre del componente | Nombre del sub-componente | Cobertura | Nuevo, Modificado, Eliminado |
|-------|---|---|----------------|------------------------------|
| 1 | Proceso de carga en la cartera de solicitudes de fibra | load_cartera_fibra.sql merge_cartera_cdt | Total Total | Modificado Nuevo |
| 2 | Cartera de solicitudes | Cartera | Total | Modificado |
| 3 | Histórico de solicitudes | Historico | Total | Modificado |
| 4 | Proceso de carga en la cartera de solicitudes de cobre | cartera_altas.sql | Total | Modificado |
| 5 | Proceso de carga en el histórico de solicitudes provenientes de cobre | historico_diario.sql | Total | Modificado |
| 6 | Proceso de carga en el histórico de solicitudes provenientes de fibra | load_historico_fibra.sql | Total | Modificado |

Tabla 5-1: Cobertura de requisitos.

5.1.2 Listado de sub-componentes

A continuación, en la Tabla 5-2, se listan los componentes y los subcomponentes a modificar o crear, junto con un resumen de los mismos.

| ID | Nombre Componente | Tipo Componente | Nombre de sub-componentes | Resumen ejecutivo subcomponentes software |
|----|--|-----------------|---|--|
| 1 | Proceso de carga de Cartera con solicitudes de fibra | Proceso | load_cartera_fibra.sql merge_cartera_cdt.sql | Modificación del proceso de carga load_cartera_fibra.sql que realiza las altas en la tabla de Cartera de las nuevas solicitudes reportadas por el provisionador de fibra. Se crea la caja merge_cartera_cdt, así como el proceso merge_cartera_cdt.sql. |
| 2 | Cartera de solicitudes | Tabla | Cartera | Modificación de la tabla Cartera. Se incluye un identificador de la solicitud origen, un indicador de la migración y fecha de la última migración de ADSL a fibra. |
| 3 | Histórico de solicitudes | Tabla | Historico | Modificación de la tabla Historico. Se incluye el identificador de la solicitud origen. |
| 4 | Proceso de carga de Cartera con solicitudes de cobre | Proceso | cartera_altas.sql | Modificación del proceso que actualiza la Cartera del cliente de acuerdo con las solicitudes de alta recibidas del provisionador de ADSL. |
| 5 | Proceso de carga de Historico con solicitudes de cobre | Proceso | historico_diario.sql | Modificación del proceso que actualiza el histórico de solicitudes, de acuerdo a las solicitudes recibidas del provisionador de cobre. |
| 6 | Proceso de carga de Historico con solicitudes de fibra | Proceso | load_historico_fibra.sql | Modificación del proceso que actualiza el histórico de solicitudes de acuerdo a las solicitudes recibidas del provisionador de fibra. |

Tabla 5-2: Listado de subcomponentes.

5.1.3 Descripción de los procesos

ID-1 Proceso de carga de la tabla Cartera con solicitudes provenientes de fibra

Capacidades funcionales iniciales

Proceso encargado de incluir las solicitudes de alta provenientes del provisionador de fibra para un contrato nuevo.

Capacidades funcionales final

Se realizará la carga de las solicitudes de alta que llegan desde el provisionador de fibra, pero solo se insertarán las que no pertenezcan a un cambio de tecnología (como ya se ha mencionado, se identifica mediante id_solicitud_padre a -1 y el tipo de solicitud a 20). Así se evita la aparición de dos registros al haber un cambio de tecnología.

Así mismo, se modificará el proceso load_cartera_fibra.sql para informar en Cartera los tres campos creados en la tabla:

- Identificador de la solicitud padre del contrato.
- Un identificador de migración de ADSL a fibra.
- Fecha de la última migración.

Resumen de las modificaciones del componente

Se modifica el proceso load_cartera_fibra.sql para que solamente inserte las solicitudes que vengan de fibra cuyo tipo de solicitud no se trate de un cambio de tecnología.

Para tratar los registros de cambios de tecnologías, no hay que insertarlos como una nueva alta en la tabla Cartera, sino que se deberán actualizar los campos del registro existente con la información llegada desde el provisionador de fibra. Para ello se crea una nueva caja merge_cartera_cdt, que será la que los trate.

Con la ejecución de las 2 cajas (preexistente load_cartera_fibra y nueva merge_cartera_cdt) se procesa la totalidad de los tipos de registro: los que no son de cambio de tecnología, y los que son de cambio de tecnología.

Así mismo, se modificará el proceso load_cartera_fibra.sql para informar en la tabla Cartera los campos creados en la tabla de Cartera. Estos campos contendrán:

- Campo id_solicitud_origen. Este registro permanecerá informado, aunque se produzcan modificaciones en el registro.
- El campo ind_migracion: se informará con la incorporación de un nuevo registro de alta proveniente desde fibra. Estará activo cuando se haya registrado una solicitud con una migración de cobre a fibra.
- Campo fe_ult_migracion: se informará con la fecha de la migración de cobre a fibra.

Id-2 Tabla de Cartera

Capacidades funcionales iniciales

Tabla donde se registra la cartera de clientes.

Capacidades funcionales final

Se crearán varios campos en los cuales se informará:

- La solicitud padre de un contrato.
- Un indicador de migración.
- La fecha de la última migración.

Resumen de las modificaciones del componente

En esta tabla se crearán los campos ya mencionados anteriormente:

- id_solicitud_origen. Identificador de la primera solicitud generada al cliente (solicitud padre del contrato).
- ind_migracion. Identificador de migración.
- fe_ult_migracion. Fecha de última migración.

Id-3 Tabla Historico, la tabla del histórico de solicitudes

Capacidades funcionales iniciales

Tabla donde se registra el histórico de solicitudes de un cliente.

Capacidades funcionales final

Se creará un campo en el que se informará la solicitud padre del contrato.

Resumen de las modificaciones del componente

En esta tabla se informará el campo id_solicitud_origen. Identificador de la primera solicitud generada al cliente.

Id-4 Proceso de carga de la tabla Cartera con solicitudes provenientes de cobre

Capacidades funcionales iniciales

Proceso encargado de incluir las solicitudes de alta provenientes de cobre (ADSL) para un contrato nuevo.

Capacidades funcionales final

Se modificará el proceso para informar en Cartera el identificador de la solicitud padre del contrato.

Resumen de las modificaciones del componente

Se modificará el proceso cartera_alta.sql para informar en la tabla Cartera el campo id_solicitud_origen, que informará lo mismo que en los componentes ya descritos. Este registro permanecerá informado, aunque se produzcan modificaciones en el registro.

ID-5 Proceso de carga de solicitudes de cobre en el histórico de solicitudes

Capacidades funcionales iniciales

Proceso encargado de incluir el histórico de solicitudes en la tabla Historico.

Capacidades funcionales final

Se modificará el proceso para informar el campo id_solicitud_origen en la tabla Historico con el identificador de la primera solicitud de alta en el registro del histórico de solicitudes.

Resumen de las modificaciones del componente

El proceso historico_diario.sql informará en el campo id_solicitud_origen el identificador de la solicitud padre del contrato.

ID-6 Proceso de carga de solicitudes de fibra en el histórico de solicitudes.

Capacidades funcionales iniciales

Proceso encargado de incluir el histórico de solicitudes en la tabla Historico.

Capacidades funcionales final

Se modificará el proceso para informar en la tabla Historico el campo id_solicitud_origen.

Resumen de las modificaciones del componente

El proceso load_historico_fibra.sql informará en el campo id_solicitud_origen el identificador de la solicitud padre del contrato.

ID-7 Universo de Business Objects

Capacidades funcionales iniciales

Alguna de la información que se puede consultar en este Universo actualmente es la siguiente:

- Cartera de clientes
- Oferta Actual
- Histórico de solicitudes
- Datos personales del cliente

Capacidades funcionales final

Se modificará el universo para poder visualizar el Motivo de Cancelación.

Resumen de las modificaciones del componente

Se modificará el universo para poder visualizar el Motivo de Cancelación que proviene directamente del provisionador de fibra.

5.1.4 Carga inicial y normalización de datos

Para el correcto funcionamiento de las nuevas funcionalidades, se debe realizar una primera carga cuya finalidad será actualizar los nuevos campos creados en las tablas Cartera e Historico, con la información correspondiente a los datos ya existentes en el Data Warehouse. Es decir, a cada registro existente en ambas tablas asignarle al campo id_solicitud_origen creado su correspondiente solicitud origen. Además, en Cartera, en caso de que el registro sea un cambio de tecnología, asignarle la fecha de migración y poner el indicador de migración a 1.

5.2 Flujogramas

R1. Unificación en un solo contrato de los cambios de provisionador, se necesita que los cambios de provisionador de cobre a fibra no supongan en los analíticos una baja y un alta del contrato. El flujograma de la solución adoptada para este requisito se puede ver en la Figura 5-1.

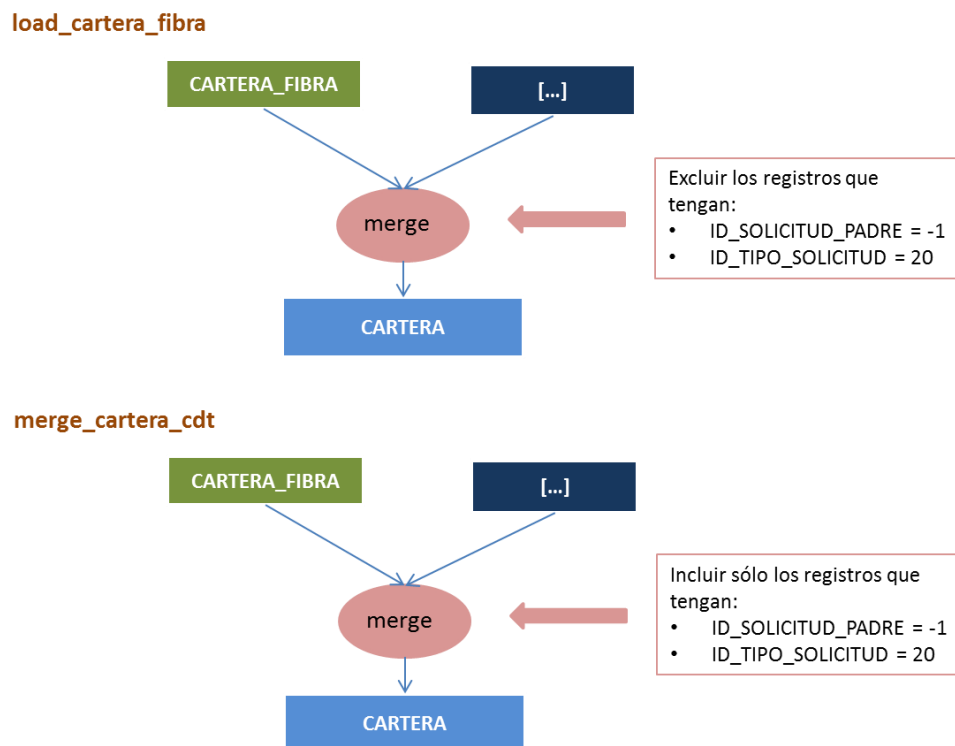


Figura 5-1: Flujograma requisito 1.

R2. Identificación en el histórico de solicitudes el cambio de tecnología de cobre a fibra. Incluimos campos para el indicador de migración y fecha ultima migración. Este requisito se representa en el flujograma de la

Figura 5-2.

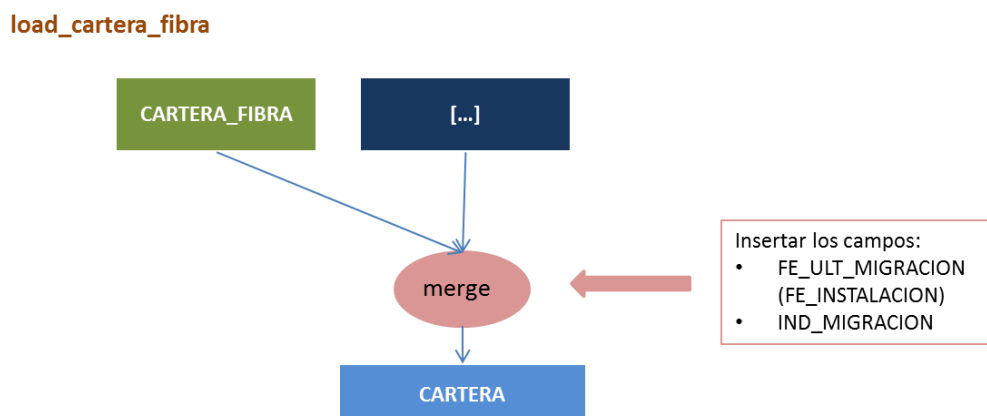


Figura 5-2: Flujograma requisito 2.

R3. Implementar una solución para la trazabilidad de las solicitudes por las que ha pasado un cliente. Incluimos el campo solicitud origen en la Cartera, el cual contendrá la primera solicitud padre del contrato (considerando todos los provisionadores). El flujograma de la solución a este requisito se representa en la Figura 5-3 para la cadena de fibra, y en la Figura 5-4 para la cadena de cobre.

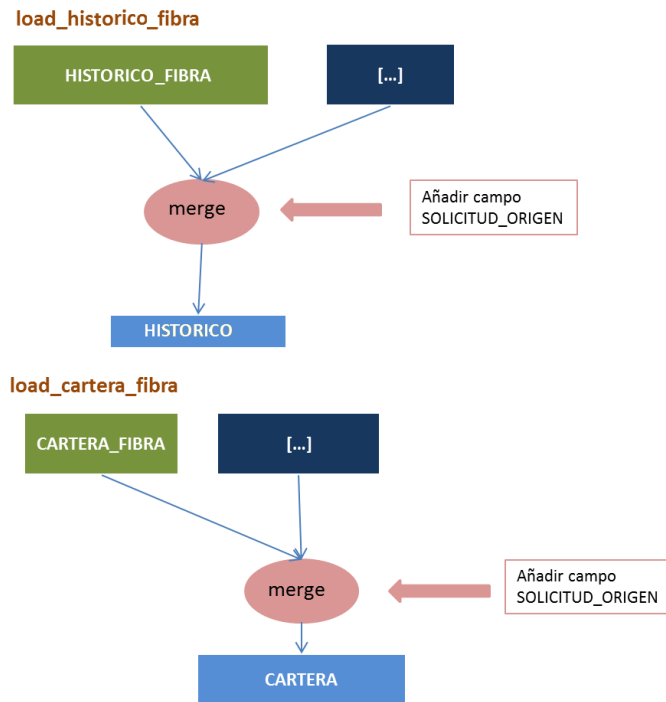


Figura 5-3: Flujograma requisito 3 cadena de fibra.

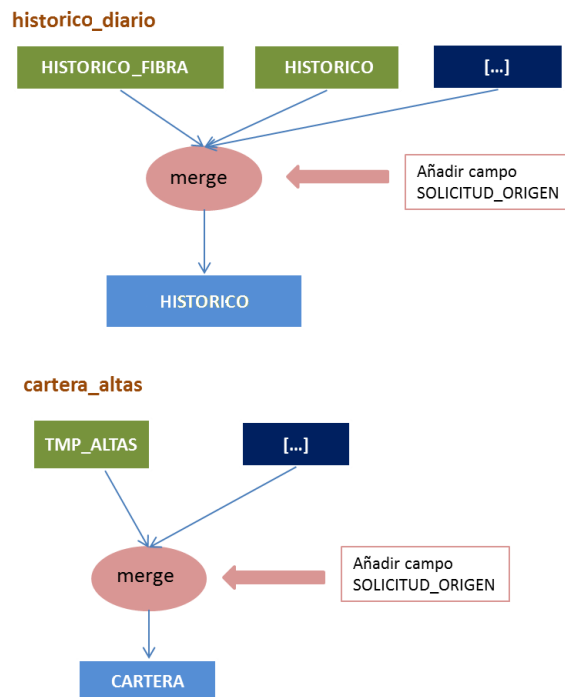


Figura 5-4: Flujograma requisito 3 cadena de cobre.

5.3 Acuerdo de interfaz

Se ha debido modificar la interfaz entre el Data Warehouse del Fijo y el Analítico del Fijo, pues el sistema fuente de éste último, es el Data Warehouse. Los acuerdos de interfaz entre los distintos sistemas se pueden observar en la Tabla 5-3 y en la Tabla 5-4, representándose en la primera el acuerdo de interfaz para la tabla de Cartera, y en la segunda

para la tabla de Historico. Además de los nuevos campos añadidos a las tablas, se mapea el campo Fe_ult_instalacion solicitado por el equipo encargado del sistema Analítico del fijo.

| Mapeo de Cartera | | | | | |
|------------------|---------------------|--------|-------------|---------------------|--------|
| Origen | | | Destino | | |
| Entidad | Campo | Tipo | Entidad | Campo | Tipo |
| Cartera | Fe_ult_instalacion | Date | Stg_cartera | Fe_ult_instalacion | Date |
| Cartera | Ind_migracion | Number | Stg_cartera | Ind_migracion | Number |
| Cartera | Fe_ult_migracion | Date | Stg_cartera | Fe_ult_migracion | Date |
| Cartera | Id_solicitud_origen | Number | Stg_cartera | Id_solicitud_origen | Number |

Tabla 5-3: Acuerdo de interfaz para la Cartera.

| Mapeo de Historico | | | | | |
|--------------------|---------------------|--------|---------------|---------------------|--------|
| Origen | | | Destino | | |
| Entidad | Campo | Tipo | Entidad | Campo | Tipo |
| Historico | Fe_ult_instalacion | Date | Stg_historico | Fe_ult_instalacion | Date |
| Historico | Id_solicitud_origen | Number | Stg_historico | Id_solicitud_origen | Number |

Tabla 5-4: Acuerdo de interfaz para el histórico de solicitudes.

5.4 Planificación de ejecución

Los procesos que componen la cadena son llamados mediante otros procesos, las cajas. La nueva caja que se ha creado se ejecuta después que haya terminado la caja load_cartera_fibra y comienza con la correcta finalización de ésta, así mismo, la caja posterior load_historico_fibra, comenzará con la correcta finalización de la nueva caja merge_cartera_cdt, como se puede observar en la

Figura 5-5.

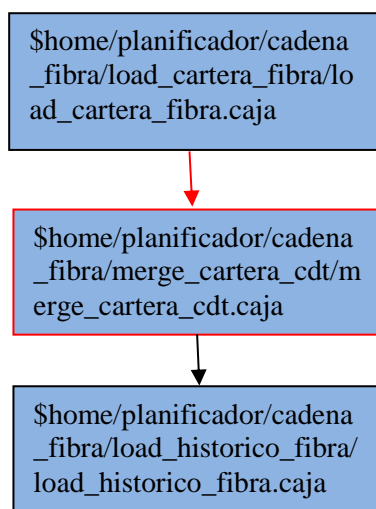


Figura 5-5: Ejecución cajas de la cadena de fibra.

6 Desarrollo

Durante la fase de implementación se ha realizado el desarrollo de los distintos componentes para satisfacer los requisitos pedidos por el cliente.

Por un lado, se han modificado los procesos ETL para la carga tanto de solicitudes provenientes de cobre como para las provenientes de fibra, en la tabla Cartera y la tabla de histórico. Estos procesos modificados se ejecutan diariamente y afectarán a las futuras solicitudes recibidas.

Así mismo, se han añadido nuevos campos a estas dos tablas. El desarrollo para ello se ha ejecutado una sola vez, y no se volverá a ejecutar, a excepción de que se necesite realizar una marcha atrás y volver a realizar el despliegue del proyecto.

Además, se ha realizado la primera carga la cual ha actualizado los nuevos campos creados en las tablas con la información correspondiente a los datos ya existentes en el Data Warehouse.

También fue necesario realizar la normalización de los datos, dejando solo un registro en Cartera en los cambios de tecnología, evitando así que haya un registro de baja y otro de alta.

Al finalizar el desarrollo, y tras un resultado positivo de las pruebas, se han realizado dos archivos de extensión .tar: uno para poder realizar el pase a producción y otro de marcha atrás. Ambos contienen una estructura de carpetas igual a la del sistema, respetando las jerarquías, para así poder realizar un despliegue satisfactorio.

Todo el desarrollo se ha realizado en el lenguaje de programación PL/SQL.

Aquí se explicarán brevemente las ideas de cómo se ha realizado el desarrollo. En el apartado de manual de instalación se explicarán los pasos a seguir para una correcta instalación o marcha atrás del código.

6.1 Instalación, carga inicial y normalización

En esta parte se realiza la modificación de las tablas para añadir los campos nuevos, se realiza una carga inicial en la cual se rellenan esos nuevos campos en los registros ya existentes con la información correspondiente, y se realiza una normalización de los datos para corregir la situación de existencia de una baja y un alta cuando ha habido un cambio de tecnología. Para ello, se han realizado los siguientes pasos:

- Creación de una tabla temporal que va a alojar el identificador de la solicitud origen. Esta tabla además de los cuatro campos identificativos, contendrá el nuevo campo identificador de la solicitud origen, el cual se obtiene a partir de la tabla histórico de solicitudes, haciendo un *group by* de esos cuatro campos, y seleccionando el mínimo identificador de solicitud. Se crea así mismo un índice para esa tabla temporal a partir de esos cuatro campos identificativos.
- Renombrar las tablas Cartera e Historico, y crear unas nuevas con los mismos datos, además de los nuevos campos a añadir. Para ello, primero realizamos un *alter table* del histórico de solicitudes, la llamamos Historico_cdt. Luego realizamos un *create table ... as select ... from* Historico_cdt, con todos los campos ya existentes, manteniendo así la información que había previamente, y además le añadimos el campo *id_solicitud_origen*, que lo cogemos de la tabla temporal creada en el apartado anterior, y buscando por los cuatro campos ya mencionados. Al crearla, la

almacenamos en uno de los *tablespace*⁵ para tablas (*data_table_**). Lo mismo para la tabla Cartera, pero además de la solicitud origen le añadimos el campo *fe_ult_migracion* a null y el indicador de migración a 0.

Se crean los sinónimos para las tablas mediante *create or replace public synonym* y se otorgan los mismos permisos que tenían las tablas originales con *grant select on*. Así mismo, se borran los índices existentes y se crean los mismos para las nuevas tablas creadas, almacenándolos en los *tablespace* para índices (*data_idx_**).

Para mejorar el rendimiento, se crean estadísticas sobre las tablas e índices creados mediante *gather_table_stats*, que es un procedimiento que permite analizar las estadísticas de las tablas, columnas e índices de manera más óptima. [26]

- Se regularizan las bajas. Se buscan en el histórico de solicitudes los clientes que hayan tenido cambio de tecnología (el identificador tipo solicitud 20 y el identificador estado solicitud 2, que es estado de solicitud completado), que provengan de tecnología ADSL, mediante la creación de un cursor. Para cada registro encontrado, se busca el registro de baja en Cartera mediante un bucle *loop*, de forma que coincidan los cuatro campos mencionados, el campo del estado del cliente sea baja y el tipo de producto sea ADSL. Si se encuentra, se borran dichos registros de la Cartera a partir del identificador de solicitud.
- Se elimina la tabla temporal creada en el primer paso.
- Durante el proceso de instalación se ha tomado la precaución de crear tablas *backups* de Cartera e Historico en caso de ser necesaria una recuperación de datos. Tras comprobar que el pase a producción se ha efectuado correctamente se deberán borrar dichas tablas *backups*. Esto se hará efectivo 4 días tras la realización del pase.

6.2 Desarrollo de componentes

6.2.1 Cadena de fibra

Proceso de carga de altas en la Cartera

El proceso *load_cartera_fibra* realiza la carga en la tabla Cartera de las altas llegadas del provisionador de fibra.

Como un cambio de tecnología actualmente se registra como una baja en cobre y un alta en fibra, interesa modificar la lógica de este proceso, pues llegará el cambio a fibra como un alta.

Dicho proceso realiza un *merge* en la tabla Cartera a partir del identificador de solicitud, cogiendo los datos de la tabla *Cartera_fibra* la cual llega del provisionador de fibra y contiene los nuevos registros a insertar o actualizar en Cartera.

Se modifica el proceso para que solamente inserte las solicitudes que sean provenientes del provisionador de fibra cuyo tipo de solicitud no se trate de un cambio de tecnología, es decir, sea propiamente un alta. Para ello, se ha modificado el archivo *load_cartera_fibra.sql*, para que en la *query select*, no coja los cambios de tecnología, los cuales como ya hemos mencionado, vienen indicados con un el identificador de la solicitud padre a -1 y el identificador del tipo de solicitud a 20.

⁵ Un espacio de tabla (*tablespace* en inglés) es una ubicación de almacenamiento donde pueden ser guardados los datos correspondientes a los objetos de una base de datos.

Así mismo, se han añadido los tres campos siguientes:

- **Id_solicitud_origen:** identificador de la primera solicitud de alta en el registro del histórico de solicitudes.
- **El campo ind_migracion:** se informará con la incorporación de un nuevo registro de alta proveniente desde fibra. Estará activo cuando se haya producido una migración de cobre a fibra.
- **Campo fe_ult_migracion:** se informará con la fecha de registro de una solicitud proveniente de fibra que sea un cambio de tecnología.

Estos campos, en caso que el registro ya exista en la Cartera (si coincide el identificador de solicitud en el *merge*) no se actualizarán. En caso de que no exista y se inserte un nuevo registro, que corresponde con un alta de un cliente, se informará el **id_solicitud_origen** con su propio número de solicitud (pues por ser un alta, es la primera solicitud asociada al cliente), y el campo **fe_ult_migracion** se pondrá a *null* y el **ind_migracion** a cero, debido a que aquí sólo se tratan solicitudes que no sean cambios de tecnología, y en este caso los valores correspondientes a estos campos son esos.

Proceso de carga de los cambios de tecnología en la Cartera

Para tratar los registros de cambios de tecnologías, no hay que insertarlos como una nueva alta, sino que se deberán actualizar los campos del registro existente en la Cartera con la información proveniente desde el provisionador de fibra.

Para ello se ha creado una nueva caja **merge_cartera_cdt**. En su correspondiente archivo SQL, hay un *merge* en el que en la función *select* solamente se cogen los registros de tipo cambio de tecnología, y se actualizan los campos debidos.

Al igual que el otro proceso, coge la información de la tabla **Cartera_fibra**, pero esta vez no compara por el identificador de solicitud, sino que como no será un alta de un cliente nuevo, sino que el cliente ya existe en la Cartera, compara por los cuatro campos identificativos del cliente (número de contrato, número de teléfono, numero de documento y tipo de documento).

Cuando coincidan, tanto si el número de solicitud es el mismo y se está realizando una actualización, o bien si es un nuevo número de solicitud y el registro de la Cartera tiene el tipo de producto ADSL y el estado del cliente Baja, se actualizarán los campos con la información proveniente del provisionador de fibra.

Cabe destacar que:

- El identificador de la solicitud origen no se modifica. No interesa modificarlo pues será el mismo que había antes del cambio de tecnología.
- El indicador de migración se pondrá a 1, al considerarse el cambio de tecnología una migración de ADSL a fibra.
- La fecha de migración se pondrá como la fecha de instalación que llega desde provisionador de fibra.
- El campo de fecha de instalación no se actualizará. Deberá mantener la fecha del contrato ADSL. El que sí se actualizará será el campo de fecha de la última instalación, la cual informará la fecha de instalación de fibra.

Con la ejecución de las dos cajas, la caja preexistente `load_cartera_fibra` y la nueva caja `merge_cartera_cdt`, se procesa la totalidad de los tipos de registro: los que no son de cambio de tecnología, y los que son de cambio de tecnología.

Como resultado de estos cambios, cuando haya un cambio de tecnología ya no habrá dos registros en la Cartera, uno de alta y uno de baja. En lugar de insertar un registro de alta con tecnología fibra, se modificará el registro existente, el cual tiene tecnología ADSL y estado baja. Ahora dicho registro indicará tecnología fibra y estado alta, además de informar los nuevos campos creados.

Ha sido necesario crear una nueva caja en vez de procesar la funcionalidad como una condición diferente dentro del proceso existente debido a que el *merge* se realiza en cada caso por distintos campos. En el proceso existente, el *merge* se realiza por el identificador de la solicitud, y en el nuevo proceso creado, por los cuatro campos ya mencionados.

Proceso de carga de altas en el histórico de solicitudes

El proceso `load_historico_fibra` carga la tabla de histórico de solicitudes a partir de los nuevos registros provenientes del provisionador de fibra. En este proceso se realiza un *merge* en la tabla del histórico de solicitudes con los registros de la tabla `Historico_fibra`, la cual proviene del provisionador de fibra, y tendrá solamente los registros a insertar o actualizar en el histórico.

El *merge* compara a partir del identificador de solicitud. Si coincide se actualizan ciertos campos, a excepción del identificador de la solicitud origen. Si la solicitud no está en el histórico, ésta se inserta con la información proveniente de fibra, y la solicitud origen será la misma que se haya insertado antes en la Cartera. Esto se puede realizar debido a que la cadena de fibra es secuencial, y ejecuta primero la caja que carga la Cartera y luego la que carga el histórico. Así, el `id_solicitud_origen` se coge de la tabla Cartera, buscando dicho registro a partir los cuatro campos identificativos del registro a insertar.

6.2.2 Cadena de cobre

Proceso de carga de altas en el histórico de solicitudes

Se modificará el proceso `historico_diario` para informar el campo `id_solicitud_origen` en la tabla `Historico`.

El proceso realiza un *merge* en la tabla `Historico` a partir de varias tablas, entre ellas el propio `Historico` y la `tmp_historico`, la cual proviene del provisionador de cobre y contiene las solicitudes de cobre a insertar o actualizar en el histórico. El *merge* lo realiza a partir del identificador de solicitud.

Si el identificador de solicitud ya existe en `Historico`, se actualizarán ciertos datos, pero la solicitud origen no se actualiza. En caso que sea una nueva solicitud y se realice un *insert*, para obtener el valor del campo `id_solicitud_origen` se realiza lo siguiente:

1. Antes de nada, se comprueba si es un alta (si el identificador de la solicitud padre es -1 y el identificador tipo solicitud es 1). En el caso que sea un alta se pone como solicitud origen su propio identificador de solicitud, la del registro de la tabla temporal de cobre, pues es un alta de un nuevo cliente y no existe una solicitud anterior a ésta.
2. En el caso que no sea un alta y que el cliente tenga solicitudes en el histórico, se coge la solicitud origen de alguna solicitud que esté en el histórico donde coincidan los cuatro campos identificativos con aquellos del registro que se está procesando.

3. Puede ocurrir que las dos condiciones anteriores no se cumplan, es decir, que no sea un alta pero que el cliente no tenga solicitudes en el histórico. Esto puede suceder debido a que, si es un cliente nuevo, el alta se podría haber realizado ese mismo día y dicho registro de alta se encontraría en la tabla temporal proveniente de cobre.

Aunque el alta se hubiese procesado antes que esta solicitud, debido a que el *commit* se realiza después de haber hecho el *merge* de todos los registros a procesar, aun así, no se encontraría el registro de alta en el histórico.

Debido a que en la tabla temporal los registros no están ordenados y no se puede suponer que llegará antes el alta que otra solicitud de ese mismo cliente, para obtener el identificador de la solicitud origen se realiza un *over (partition by)*. El *over (partition by)* se realiza por los cuatro campos identificativos sobre la tabla temporal del histórico que llega de cobre, y se coge el mínimo identificador de solicitud de todas las solicitudes que tenga este cliente en la tabla temporal (comparando por esos cuatro campos el registro de la tabla temporal que estamos insertando con todos los demás ya procesados o por procesar de la misma tabla temporal).

Proceso de carga de altas en la Cartera

Se modificará el proceso *cartera_alta*s. El proceso consiste en un *merge* en Cartera, cogiendo información de una variedad de tablas, entre ellas *tmp_alta*s, la cual contiene las solicitudes a insertar o actualizar en Cartera provenientes del provisionador de cobre.

En el caso en que se vaya a realizar un *insert*, es decir, sea una nueva alta de cobre, se modifica el proceso para que informe los nuevos campos creados en la tabla Cartera. Como es un alta de un nuevo cliente no existente previamente, el identificador de la solicitud origen será el identificador de solicitud de él mismo. El campo indicador de migración y fecha de migración estarán a 0 y *null* respectivamente, pues es un alta en cobre, y no hay una migración en este caso.

En el caso que se realice un *update* sobre ciertos campos, estos últimos tres mencionados no se actualizarán.

6.3 Marcha Atrás

En el caso que durante el pase a producción suceda algún error, se puede realizar la marcha atrás que deja el sistema en su estado previo.

Para ello, se hace un *alter table* de la tabla histórico de solicitud y se renombra a *Historico_cdt2*, y la tabla *Historico_cdt* (que es la temporal que se ha creado de *backup*) a *Historico*. Igual con Cartera. Dejando así las tablas de *backups* como las nuevas tablas en funcionamiento. Igual que antes, se crean sinónimos y permisos para las nuevas tablas, se borran los índices existentes y se crean los mismos para la nueva tabla.

Mediante un script se borran las carpetas creadas para la nueva caja *merge_cartera_cdt*: la correspondiente al log, la que alberga el fichero SQL, la que alberga el script correspondiente, la del archivo de configuración correspondiente, y la del *.caja*. Dejando así el sistema en su estado anterior al pase a producción.

6.4 Universo Business Objects

Se solicitaba modificar el universo del sistema Data Warehouse para poder visualizar el campo Motivo de Cancelación de un contrato.

El motivo de cancelación proviene directamente del proveedor de fibra.

En el universo, de toda la información que se puede consultar, está el histórico de ofertas. Dentro del histórico de ofertas, están las cancelaciones de fibra. Desde el Business Objects Designer, se ha incluido dentro de la clase Cancelaciones Fibra (la cual está dentro de la clase Histórico Ofertas), la dimensión Motivo Cancelación, la cual realiza una *select* del campo motivo_cancela a la vista proveniente de fibra, v_fibra_cancela.

7 Integración, pruebas y resultados

En la fase de pruebas, se ha realizado el *Test Plan* y el *Test Report*, así como una estimación de base de datos.

Al ejecutar en la máquina de Unix una caja, se ejecuta toda la funcionalidad relativa a dicha caja, pero las cadenas se ejecutan en la máquina de Unix del entorno de producción a través de Control-M, una herramienta de planificación y automatización de tareas [27]. Como no se dispone de dicha herramienta ni en el entorno de desarrollo ni en el de integración, para las pruebas se ha desarrollado un pequeño programa que ejecuta una cadena y genera un log con la información de la ejecución de ésta.

Las pruebas se han realizado primero a nivel de caja, y una vez verificado el correcto funcionamiento de éstas, se ha realizado a nivel de cadena. Así mismo, primero se han probado en el entorno de desarrollo, y una vez que funcionaba allí se probaba en el entorno de integración.

El desarrollo del fichero .sql se podía probar directamente desde el PL/SQL Developer, conectándose mediante éste a la base de datos del entorno de desarrollo o integración, pero para ejecutar tanto la caja como la cadena, se debía acceder en remoto a las máquinas Unix, y lanzar la caja o la cadena directamente mediante línea de comandos.

Así mismo, fue necesaria la preparación de datos para las pruebas en los distintos entornos.

Debido a la extensión de las pruebas, se expondrán aquí sólo las más relevantes.

7.1 Pruebas de despliegue

Las pruebas consisten en la creación de los nuevos campos en las tablas Cartera e Historico y la inserción de datos en las mismas. Aquí se explicará lo principal a nivel de pruebas, para más detalle del proceso remitirse al apartado 6.1 Instalación, carga inicial y normalización.

La precondition será que el archivo dwh_fijo.tar se encuentre en el directorio \$home. Se describen a continuación los pasos a realizar y los resultados esperados:

1. Desempaquetar el archivo en la ruta \$home ejecutando el siguiente comando:
`tar -pxvf dwh_fijo_cdt.tar`
Resultado: descompresión de los directorios correspondientes.
2. Creación de directorios e inserción de los archivos.
Ejecutar el archivo dwh_fijo_crea_carpetas.sh y comprobar el acceso a la ruta \$home/planificador/fibra/merge_cartera_cdt y \$home/log/fibra/merge_cartera_cdt.
Resultado: directorios y archivos creados en el sistema.
3. Creación de la tabla tmp_carga_solicitud. Se accede a la ruta ./instalar y se ejecuta *sqlplus*, y allí el fichero dwh_fijo_create_tmp_carga_solicitud.sql
Resultado: tabla temporal creada y en ella se aloja el nuevo campo id_solicitud_origen calculado como la mínima id_solicitud para registros con mismo id_contrato, id_telefono, id_num_documento, id_tipo_documento dentro de la tabla Historico.

```

--> @DWH_FIJO_CREATE_TMP_CARGA_SOLICITUD.sql

Table created.

Index created.

SQL>

```

Figura 7-1: Pruebas de despliegue, creación tabla temporal.

En la Figura 7-1 se puede observar el resultado esperado tras la ejecución de los pasos anteriores, y en la Figura 7-2 se puede observar la correcta creación de la nueva tabla tmp_carga_solicitud (se ha contrastado con los datos de la tabla Historico).

| ID_TELEFONO | ID_TIPO_DOCUMENTO | ID_NUM_DOCUMENTO | ID_CONTRATO | ID_SOLICITUD_ORIGEN |
|-------------|-------------------|------------------|-------------|---------------------|
| 918829772 | 1 | 08868786Z | 10243696991 | 202368220 |
| 884808682 | 1 | 53788839R | 39969682024 | 2020540466 |
| 968042888 | 1 | 04686807Z | 39859773699 | 9410277252 |
| 918821291 | 1 | 70304951Z | 19310246991 | 109413666 |

Figura 7-2: Pruebas de despliegue, datos tabla temporal.

- Renombrar las tablas Cartera e Historico como Cartera_cdt y Historico_cdt. Crear nuevamente las tablas Cartera y Historico con los campos id_solicitud_origen, fe_ult_migracion e ind_migracion en Cartera e id_solicitud_origen en Historico. Para ello, se ejecuta:

```

dwh_fijo_crea_campos_historico.sql
dwh_fijo_crea_indice_historico.sql
dwh_fijo_analiza_historico.sql
dwh_fijo_crea_campos_cartera.sql
dwh_fijo_crea_indice_cartera.sql
dwh_fijo_analiza_cartera.sql

```

Resultado: renombramiento de tablas. Creación de nuevas tablas con los campos nuevos. En la Figura 7-3 se refleja el resultado tras la ejecución de dwh_fijo_crea_campos_historico.sql. Para Cartera el resultado esperado es el mismo.

```

SQL> @DWH_FIJO_crea_campos_HISTORICO.sql

Table altered.

Table created.

Synonym created.

Grant succeeded.

Grant succeeded.

Grant succeeded.

Grant succeeded.

SQL>

```

Figura 7-3: Pruebas de despliegue, creación de campos.

5. Normalización de los datos. Se ejecuta el archivo
dwh_fijo_regulariza_bajas_cartera.sql
Resultado: los clientes que tenían un registro de baja y otro de alta en Cartera pasan a tener sólo un registro.
6. Eliminación de la tabla tmp_carga_Historico.
Se ejecuta el fichero dwh_fijo_drop_tmp_carga_solicitud.sql
Resultado: tabla temporal eliminada.

7.2 Pruebas de marcha-atrás

Se probará la marcha atrás del proceso de creación del nuevo campo en las tablas Cartera e Historico y la inserción de datos en las mismas. Como precondition, el archivo dwh_fijo_cdt_ma.tar deberá estar en el directorio \$home.

Pasos a realizar:

1. Descomprimir dwh_fijo_cdt_ma.tar.
Resultado: descompresión de los directorios correspondientes.
2. Borrado de las tablas Cartera e Historico. Renombrado de las tablas Cartera_cdt y Historico_cdt a Cartera e Historico respectivamente.
Resultado: tablas borradas y renombradas.
3. Eliminación de los directorios creados.
Resultado: directorios y archivos borrados del sistema.

7.3 Pruebas progresivas

Se han realizado pruebas unitarias y de integración de las distintas funcionalidades. Aquí se ilustrará el resultado de la prueba más representativa de este proyecto.

Situación de partida modificada con este evolutivo: en la tabla Cartera, tal y como se muestra en la Figura 7-4, para un cambio de tecnología se registraban dos registros:

- Baja en cobre.
- Alta en fibra (cambio de tecnología)

| ID_SOLICITUD | ID_TIPO_DOCUMENTO | ID_NUM_DOCUMENTO | ID_TELEFONO | ID_TIPO_SOLICITUD | DE_ESTADO_CLIENTE | ID_CONTRATO |
|--------------|-------------------|------------------|-------------|-------------------|-------------------|-------------|
| 5050110665 | 1 | 01284655Z | 912447424 | 20 | Activo | 1037218383 |
| 9893616 | 1 | 01284655Z | 912447424 | 1 | Baja | 1037218383 |

Figura 7-4: Pruebas progresivas, situación actual Cartera.

En el histórico de solicitudes, se tienen todas las situaciones por las que ha ido pasado el cliente, como se puede observar en la Figura 7-5.

| ID_SOLICITUD | ID_TIPO_DOCUMENTO | ID_NUM_DOCUMENTO | ID_TELEFONO | ID_TIPO_SOLICITUD | ID_ESTADO_SOLICITUD | ID_CONTRATO |
|--------------|-------------------|------------------|-------------|-------------------|---------------------|-------------|
| 1564919 | 1 | 01284655Z | 912447424 | 20 | 2 | 1037218383 |
| 1013177 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 0898567 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 0123762 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 6962564 | 1 | 01284655Z | 912447424 | 4 | 2 | 1037218383 |
| 5050110665 | 1 | 01284655Z | 912447424 | 20 | 2 | 1037218383 |
| 4819589 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 0969523 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 899296 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 174832 | 1 | 01284655Z | 912447424 | 2 | 2 | 1037218383 |
| 120780 | 1 | 01284655Z | 912447424 | 2 | 2 | 1037218383 |
| 424263 | 1 | 01284655Z | 912447424 | 2 | 3 | 1037218383 |
| 9893616 | 1 | 01284655Z | 912447424 | 1 | 2 | 1037218383 |

Figura 7-5: Pruebas progresivas, situación actual histórico.

Desarrollo cambio de tecnología: tras ejecutar los cambios desarrollados para este proyecto, en Cartera tenemos un único registro correspondiente al cambio de tecnología (alta en fibra). Además, aparecen informados los nuevos campos id_solicitud_origen, ind_migracion y fe_ult_migracion. Esto se ve reflejado en la Figura 7-6.

| ID_SOLICITUD | ID_TIPO_DOCUMENTO | ID_NUM_DOCUMENTO | ID_TELEFONO | ID_TIPO_SOLICITUD | DE_ESTADO_CLIENTE | ID_CONTRATO | ID_SOLICITUD_ORIGEN | IND_MIGRACION | FE_ULT_MIGRACION |
|--------------|-------------------|------------------|-------------|-------------------|-------------------|-------------|---------------------|---------------|-----------------------|
| 5050110665 | 1 | 01284655Z | 912447424 | 20 | Activo | 1037218383 | 120780 | | 1/22/04/2016 14:21:59 |

Figura 7-6: Pruebas progresivas, resultado en Cartera.

Como se puede ver en la Figura 7-7, el histórico no ha visto modificada su funcionalidad, pero si el nuevo campo con el id_solicitud_origen.

| ID_SOLICITUD | ID_TIPO_DOCUMENTO | ID_NUM_DOCUMENTO | ID_TELEFONO | ID_TIPO_SOLICITUD | ID_ESTADO_SOLICITUD | ID_CONTRATO | ID_SOLICITUD_ORIGEN |
|--------------|-------------------|------------------|-------------|-------------------|---------------------|-------------|---------------------|
| 1 | 1564919 | 1 | 01284655Z | 912447424 | 20 | 2 | 1037218383 |
| 2 | 1013177 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 3 | 0898567 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 4 | 0123762 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 5 | 6962564 | 1 | 01284655Z | 912447424 | 4 | 2 | 1037218383 |
| 6 | 5050110665 | 1 | 01284655Z | 912447424 | 20 | 2 | 1037218383 |
| 7 | 4819589 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 8 | 0969523 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 9 | 899296 | 1 | 01284655Z | 912447424 | 3 | 2 | 1037218383 |
| 10 | 174832 | 1 | 01284655Z | 912447424 | 2 | 2 | 1037218383 |
| 11 | 120780 | 1 | 01284655Z | 912447424 | 2 | 2 | 1037218383 |
| 12 | 424263 | 1 | 01284655Z | 912447424 | 2 | 3 | 1037218383 |
| 13 | 9893616 | 1 | 01284655Z | 912447424 | 1 | 2 | 1037218383 |

Figura 7-7: Pruebas progresivas, resultado en el histórico.

7.4 Pruebas regresivas

Una vez se hayan realizado las pruebas unitarias, se revisará en las pruebas de regresión que el modelo no ha sufrido ninguna modificación al incorporar esta nueva funcionalidad.

Las pruebas realizadas han sido:

1. La caja modificada load_historico_fibra sigue realizando correctamente su funcionalidad anterior. Para ello, se buscan registros en Historico con el mismo id_solicitud que en Historico_fibra (que como ya mencionamos es una tabla que proviene del provisionador de fibra), pero con algún campo diferente. Se ejecuta la caja, y se consulta Historico, revisando que se han actualizado todos los campos con los valores de los campos de Historico_fibra. Así mismo se buscan registros en Historico_fibra cuyo id_solicitud no se encuentre en Historico, y al ejecutar la caja dicho registro se añade a Historico.
2. La caja modificada load_cartera_fibra sigue realizando correctamente su funcionalidad. Se buscan registros en Cartera_fibra (tabla proveniente del provisionador de fibra) que no sean cambio de tecnología (id_tipo_solicitud =20 e id_solicitud_padre =-1). Se buscan dichos registros en Cartera por id_solicitud, tal que algún campo sea distinto (para luego comprobar que se actualiza). Se ejecuta la caja y se comprueba que se han actualizado en la tabla Cartera los registros con los campos provenientes de Cartera_fibra. Al igual que en la prueba anterior, se buscan registros cuyo id_solicitud esté en Cartera_fibra y no en Cartera y se comprueba que se insertan en la tabla Cartera.
3. La caja merge_cartera_cdt trata los cambios de tecnología de forma correcta, actualizando el resto de campos de igual forma que lo hacia load_cartera_fibra. Se encuentran registros en Cartera_fibra que sean de cambio de tecnología, para los cuales hay registros en Cartera, con algún campo distinto. Se ejecuta la caja, y se comprueba que se han actualizado los registros de Cartera con los datos provenientes de Cartera_fibra.
4. Se realiza lo mismo que en las pruebas anteriores, ejecutando la caja carteralta, y comprobando la actualización correcta de campos, con la información proveniente de tmp_alta (tabla proveniente del provisionador de cobre), así como la inserción de registros en la tabla Cartera.
5. Nuevamente, se repite la prueba con la caja historico_diario, comprobando con la tabla de entrada tmp_historico (tabla del provisionador de cobre) y con Historico.
6. Una vez asegurado el correcto funcionamiento de las cajas modificadas respecto a su funcionalidad anterior, se han ejecutado las cadenas principales diarias para asegurar que no haya conflicto entre la ejecución de otros procesos y las modificaciones. Tras las pruebas, el funcionamiento ha sido el esperado.

7.5 Pruebas de rendimiento

Se han ejecutado las cajas y las cadenas originales (antes de la modificación) y las nuevas modificadas, y se ha realizado una comparación y estudio de los tiempos de ejecución. Tras la realización de éstas, se ha comprobado que el tiempo de ejecución no se ha visto penalizado tras los cambios.

1. Se ha ejecutado repetidamente la caja modificada `load_cartera_fibra`, habiendo en la tabla de entrada `Cartera_fibra` registros que son de alta que no se encuentran en la tabla `Cartera`, así como otros que no son de alta. De la misma forma, se ha ejecutado la caja original `load_cartera_fibra`, y se han comparado los tiempos de ejecución. Ha habido una disminución en el tiempo de ejecución con la nueva caja. Esto era de esperar debido a que, al modificarla, aunque se ha añadido una comparación, se ha impuesto que no se traten los cambios de tecnología, por lo que el número de registros a tratar es menor que anteriormente. Una prueba más adecuada es comparar los tiempos de ejecución de la caja anterior, con la de la ejecución de las dos cajas, la modificada y la `merge_cartera_cdt`, que es la que realiza los cambios de tecnología, y con ambas cajas se tratan todos los casos que trataba anteriormente la caja `load_historico_fibra`. Realizando esto, ya no hay una disminución en el tiempo de ejecución, sino un aumento, pero no es significativo.
2. Se ha realizado la misma prueba individualmente sobre las cajas modificadas `load_historico_fibra`, `cartera_altas` e `historico_diario`, habiendo en las tablas de entrada correspondientes a cada caja datos como los ya mencionados, de alta que no se encuentran en la tabla `Cartera` o `Historico`, y otros que no son de alta y se actualizarán en el *merge*. Se ha realizado lo mismo con las cajas originales y se han comparado los tiempos. Ha habido un aumento en el tiempo de ejecución, pero la diferencia entre la duración de ambas ejecuciones no implica un retraso significativo en el proceso.
3. Se ha ejecutado repetidamente la cadena de cobre con los archivos modificados, habiendo en las tablas de entrada `tmp_historico` y `tmp_cartera` registros de alta y otros que no son de alta. Se ha ejecutado la cadena de cobre con los archivos sin modificar (los originales), habiendo en las tablas de entrada datos como los ya mencionados. Se han comparado los tiempos de ejecución de ambas cadenas. En media, el tiempo de ejecución de la cadena no se ha visto penalizada por las modificaciones.
4. Se ha realizado la misma prueba sobre la cadena de fibra, habiendo datos como los ya mencionados en las tablas de entrada `Cartera_fibra` e `Historico_fibra`. Los tiempos de ejecución de la cadena modificada no han divergido mucho de los de la cadena original.

8 Conclusiones y trabajo futuro

8.1 Conclusiones

Este proyecto tenía como objetivo diseñar e implementar una solución para mejorar un sistema analítico ya existente.

En el mundo laboral en los proyectos de mantenimiento hay diferentes líneas de servicios: correctivos, soporte, evolutivos etc. Estos se basan en sistemas ya en funcionamiento, para ambos la labor más importante es el análisis y el conocimiento de los sistemas, para a partir de ahí, poder realizar mejoras en caso de los evolutivos, y poder corregir incidencias en el caso de los proyectos de mantenimiento. Esto último también se realiza en el caso de los evolutivos después de la implantación (después de lo que se denomina el pase a producción).

Al haber sido partícipe de este proyecto desde los inicios del mismo, se ha podido experimentar en primera persona el ciclo de vida completo de un proyecto de tipo evolutivo. Y no sólo eso, sino que un tipo de proyecto aún más especial: un cambio de proveedor, pues este proyecto se ha heredado de otra empresa. Y eso se traduce en bastantes problemas por falta de conocimiento debido a que el sistema ha sido implementado por la empresa anterior y nadie en la empresa actual conoce los sistemas para poder dar apoyo.

Se han investigado los sistemas existentes para entender su estructura y funcionamiento, lo cual tanto por falta de documentación o por documentación no actualizada, ha llevado bastante esfuerzo. Una vez que se han entendido los sistemas, se ha realizado un análisis de lo que quería el cliente, y se ha diseñado una solución. Sin duda, estas dos fases son lo que más tiempo han consumido. Una vez diseñada la solución, se ha empezado a implementar, y a medida que se han ido realizando las pruebas unitarias se han ido descubriendo nuevas características que no se conocían del sistema y que invalidaban la solución propuesta, otra vez, por falta de conocimiento y buena documentación. Por lo que ha sido necesario volver a diseñar una solución, implementarla y volver a probarla. La metodología en V ha sido muy adecuada para este tipo de proyecto. Una vez implementada la solución y probada, se ha realizado el pase a producción y se han puesto en funcionamiento los cambios realizados.

Como ejemplo de uno de los problemas encontrados, se había planteado la solución de la obtención del identificador de la solicitud origen en la cadena de cobre en uno de los procesos ETL de la siguiente manera: como en la documentación no figuraba información sobre el orden de ejecución de los procesos en la cadena de cobre, y la ejecución de la cadena de fibra era en orden secuencial, se ha supuesto que la cadena de cobre también era secuencial. Se ha buscado en el entorno de producción el orden de ejecución de las cajas de esta cadena de la noche anterior (pues no se tiene acceso al planificador de tareas de producción). Como primero se ejecutaba el proceso relativo a la carga de Cartera, y luego el del histórico, para obtener el identificador de la solicitud origen en el histórico se cogía este dato de la Cartera, así la lógica y el proceso de obtención del mismo se realizaba una sola vez, en el proceso de carga de la tabla Cartera, disminuyendo así el impacto de los cambios en el rendimiento del sistema. El problema surgió al observar otro día la ejecución de la noche anterior y ver que se ejecutaba primero la caja de actualización del histórico y luego la de la Cartera. Hubo que cambiar por completo la lógica de la solución, pues se ejecutaban las cajas de la cadena de cobre en paralelo.

Del hecho de haber sido un proyecto de migración, se encontró también el problema de que había inconsistencia de datos en los sistemas, como identificadores a -1, lo que dificultaba la identificación unívoca de un cliente. Así mismo, hubo que conciliar el código con la otra empresa, la cual aún tenía proyectos en desarrollo y realizaba modificaciones en los sistemas. Fue necesario integrar el código, así como la realización de un inventario de fuentes modificados para la coordinación con la otra empresa.

Como se ha mencionado anteriormente, se ha trabajado en diferentes entornos, el de desarrollo, el de integración y el de producción. El acceso al entorno de producción ha permitido realizar mejor el análisis al poder ver los *logs* tras la ejecución de los diferentes procesos, así mismo ha sido indispensable para la realización de la estimación de bases de datos, y por supuesto, para la obtención del código de los distintos procesos y de los datos con los que testar en los entornos de desarrollo e integración.

Dicho esto, cabe mencionar uno de los problemas encontrados por causa de los entornos. Los entornos no eran exactamente iguales, en cada uno había unos *tablespaces* y unos roles disponibles diferentes a los de los otros entornos. Los que se utilizaban en uno, podían no existir o no tener espacio suficiente (en caso de los *tablespaces*) en otro entorno. Esto provocó que, al probar el código desarrollado, en el cual se habían puesto los *tablespaces* y roles de producción, éste no funcionara en los entornos de desarrollo e integración. Por lo que hubo que cambiar el código que se probó, del que se implantó.

Otro problema con los entornos surgió del hecho de trabajar con un equipo en Murcia, y por falta de coordinación al comienzo del proyecto. Al estar ellos realizando sus pruebas en el mismo entorno, borraron accidentalmente una tabla del entorno de desarrollo. Así mismo, insertaron datos que ya estaban en el entorno, provocando duplicados, y causando fallos en la ejecución de las cajas. Eso provocó la necesidad de realizar reuniones semanales de coordinación.

En definitiva, ha sido un trabajo muy enriquecedor, pues además de haber realizado todas las fases del ciclo de vida de un proyecto de software, se ha aprendido a afrontar los problemas que surgen en un proyecto real.

8.2 Trabajo futuro

En un futuro, la compañía se plantea migrar todo el sistema del Data Warehouse del Fijo, al otro sistema, el Analítico del Fijo.

Esto es muy costoso, y por ahora la operadora no tiene presupuesto suficiente para un cambio de tal envergadura.

En un futuro cercano, se realizarán dos proyectos muy similares a éste. Ambos consistirán en adaptar el Data Warehouse, uno de ellos para permitir un cambio de titular, y el otro un cambio de domicilio. Para ambos cambios se quiere un nuevo tipo de operación, traslado de domicilio en el primer caso, y cambio de titular en el segundo, que refleje en los sistemas analíticos este nuevo tipo de orden, y permita mantener el número de teléfono.

Referencias

- [1] OLTP vs. OLAP. <http://datawarehouse4u.info/OLTP-vs-OLAP.html>. Fecha de consulta: 15 de enero de 2016.
- [2] Robert Wrembel and Christian Koncilia, “Data Warehouses and OLAP: Concepts, Architectures, and Solutions”.
- [3] Oliver Camp, Joaquim B.L. Filipe, Slimane Hammoudi, Mario G. Piattini, “Enterprise Information Systems V”.
- [4] Sinnexus. Business Intelligence, Informática Estratégica. http://www.sinnexus.com/business_intelligence/olap_vs_oltp.aspx. Fecha de acceso: 10/03/2016.
- [5] Dataprix. Knowledge is the goal. <http://www.dataprix.com/blogs/respinosamilla/base-datos-anal-tica-datawarehouse-o-almac-n-datos>. Fecha de acceso: 01/06/2016
- [6] Altek Solutions. <http://alteksolutions.com/wp/index.php/2009/03/what-is-a-business-objects-universe/>. Fecha de consulta: 11/05/2016.
- [7] Todo BI. <Business Intelligence>. <http://todobi.blogspot.com.es/2006/03/business-objects-la-solucion-bi-que.html>. Fecha de acceso: 13/05/2016.
- [8] Oracle. www.oracle.com. Fecha de acceso: 10/02/2016.
- [9] Business Objects. <http://go.sap.com/solution/platform-technology/analytics/business-intelligence-bi.html>. Fecha de acceso: 10/02/2016.
- [10] Teradata. www.teradata.com. Fecha de acceso: 10/02/2016.
- [11] SAS. www.sas.com. Fecha de acceso: 10/02/2016.
- [12] ITWeb Enterprise. http://www.itweb.co.za/index.php?option=com_content&view=article&id=96120. Fecha de acceso: 10/04/2016
- [13] Laboratorio Nacional de Calidad del Software de INTECO. “Ingeniería del software: metodologías y ciclos de vida”. Marzo 2009.
- [14] William E. Lewis, “Software Testing and Continuous Quality Improvement”, Third Edition.
- [15] Oracle. <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index-092322.html>. Fecha de acceso: 20/02/2016.
- [16] Cisco VPN Client. <http://www.cisco.com/c/en/us/support/security/vpn-client/tsd-products-support-general-information.html>. Fecha de consulta: 10/02/2016
- [17] Cisco Network Admission Control. <http://www.cisco.com/c/en/us/solutions/enterprise-networks/network-admission-control/index.html>. Fecha de acceso: 10/02/2016
- [18] PL/SQL Developer. <https://www.allroundautomations.com/>. Fecha de acceso: 10/02/2016
- [19] Putty. <http://www.putty.org/>. Fecha de acceso: 10/02/2016.
- [20] Filezilla. <https://filezilla-project.org/>. Fecha de acceso: 10/02/2016.
- [21] Oracle Help Centre. https://docs.oracle.com/cd/B28359_01/network.111/b28317/tnsnames.htm. Fecha de acceso: 20/02/2016.
- [22] Oracle Help Centre. https://docs.oracle.com/cd/B19306_01/server.102/b14357/qstart.htm. Fecha de acceso: 15/03/2016.
- [23] SVN. <https://subversion.apache.org/>. Fecha de acceso: 10/02/2016.
- [24] TortoiseSVN. <https://tortoisesvn.net/>. Fecha de acceso: 10/02/2016.
- [25] Jira. <https://es.atlassian.com/software/jira>. Fecha de acceso: 10/02/2016.
- [26] Oracle Help Centre. https://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_stats.htm. Fecha de acceso: 25/05/2016.
- [27] BMC Software. <http://www.bmcsoftware.es/it-solutions/control-m.html>. Fecha de acceso: 15/05/2016

Glosario

Data Warehouse: Almacén de datos.

Cadena: Proceso batch de ejecución planificada y automatizada. Engloba uno o más pasos individuales, que en este contexto se llaman cajas.

Caja: *Link* de Unix que hace referencia a un proceso concreto (Ej.: proceso1.caja) y se tiene a su vez un *link* al fichero proceso1.sh.

Cisco: Cisco Systems es una empresa global principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.

Fichero de configuración: Fichero que contiene variables de entorno que podrán ser utilizadas por los procesos de carga.

Fibra (óptica): La fibra óptica es un medio de transmisión, empleado habitualmente en redes de datos y telecomunicaciones, consiste en un hilo muy fino de material transparente, vidrio o materiales plásticos, por el que se envían pulsos de luz que representan los datos a transmitir.

Log: Registro. En este TFG, se refiere al mensaje que genera la ejecución de una caja, en virtud del cual se muestra información sobre cómo ha ido dicha ejecución.

Metadatos: Literalmente «sobre datos», son datos que describen otros datos. En este TFG se refiere a tablas de metadatos que almacenan constantes y variables que son utilizadas como variables de entorno por las distintas cajas.

OLAP: On-Line Analytical Processing. Los sistemas OLAP son bases de datos orientadas al procesamiento analítico en línea.

Oracle: Oracle Corporation es una compañía de software que es célebre principalmente por desarrollar bases de datos y sistemas de gestión de bases de datos, así como mucho otro software orientado al segmento empresarial. Aquí se refiere al gestor de base de datos Oracle.

Provisionador: Sistema que se encarga de hacer la provisión de un servicio o producto.

SAS: Statistical Analysis System.

SVN: Apache Subversion.

Teradata: Empresa especializada en herramientas de data warehousing y herramientas analíticas empresariales. En este TFG cuando hablemos de Teradata se refiere a la base de datos de Teradata, que es un motor de bases de datos relacional orientado al procesamiento de grandes volúmenes de datos.

Test plan: Plan de pruebas. Un documento que describe los objetivos y actividades de la fase de pruebas.

Test report: Informes de pruebas.

Universo Business Objects: Capa semántica entre la base de datos de una organización y el usuario final.

UNIX: Es un sistema operativo portable, multitarea y multiusuario, popular en entornos cliente/servidor empresariales, debido a su estabilidad y popularidad.

Acrónimos

| | |
|-----------------|--|
| ADSL | Asymmetric Digital Subscriber Line |
| BI | Business Intelligence |
| BO | Business Objects |
| DWH | Data Warehouse |
| ETL | Proceso de Extracción, Transformación y Carga de datos |
| FTP | File Transfer Protocol |
| FTTH | Fiber To The Home |
| IP | Internet Protocol |
| PL/SQL | Procedural Language/Structured Query Language |
| TERADATA | Teradata Database Software |
| TFG | Trabajo de Fin de Grado |
| VPN | Virtual Private Network |

Anexos

A Manual de despliegue

Se ha documentado como se debe realizar el pase a producción y como se debe realizar la marcha atrás en caso de necesidad para el equipo de servicio. A continuación se detallan los archivos necesarios: en la Tabla 0-1 se pueden observar los archivos de instalación y de ejecución, en la Tabla 0-2 los de configuración, en la Tabla 0-3 el código fuente, y finalmente en la Tabla 0-4 los archivos de marcha atrás.

Archivos de instalación y ejecución

| N. | Nombre (ruta y nombre) | Checksum |
|----|---|----------|
| 1 | ./instalar/dwh_fijo_create_tmp_carga_solicitud.sql | 14471 |
| 2 | ./instalar/dwh_fijo_drop_tmp_carga_solicitud.sql | 08095 |
| 3 | ./instalar/dwh_fijo_crea_campos_cartera.sql | 10248 |
| 4 | ./instalar/dwh_fijo_crea_campos_historico.sql | 14309 |
| 5 | ./instalar/dwh_fijo_crea_indice_historico.sql | 37927 |
| 6 | ./instalar/dwh_fijo_crea_indices_cartera.sql | 64398 |
| 7 | ./instalar/dwh_fijo_analiza_cartera.sql | 27868 |
| 8 | ./instalar/dwh_fijo_analiza_historico.sql | 08893 |
| 9 | ./dw_planificador/cadena_fibra/merge_cartera_cdt/merge_cartera_cdt.caja | 10646 |
| 10 | ./dw_scripts/cadena_fibra/merge_cartera_cdt/merge_cartera_cdt.sh | 13183 |
| 11 | ./instalar/dwh_fijo_crea_carpetas.sh | 11123 |
| 12 | ./instalar/dwh_fijo_regulariza_bajas_cartera_cdt.sql | 01395 |
| 13 | ./instalar/dwh_fijo_drop_cartera_cdt.sql | 36951 |
| 14 | ./instalar/dwh_fijo_drop_historico_cdt.sql | 03805 |

Tabla 0-1: Archivos de instalación y ejecución.

Archivos de configuración

| N. | Nombre (ruta y nombre) | Checksum |
|----|---|----------|
| 1 | ./dw_cfg/cadena_fibra/merge_cartera_cdt.cfg | 29744 |

Tabla 0-2: Archivos de configuración.

Código fuente

| N. | Nombre (ruta y nombre de archivo) | Checksum |
|----|---|----------|
| 1 | ./dw_sql/cadena_cobre/cartera_altas/cartera_altas.sql | 21117 |
| 2 | ./dw_sql/cadena_cobre/historico_diario/historico_diario.sql | 00598 |
| 3 | ./dw_sql/cadena_fibra/merge_cartera_cdt/merge_cartera_cdt.sql | 53800 |
| 4 | ./dw_sql/cadena_fibra/load_cartera_fibra/load_cartera_fibra.sql | 42443 |
| 5 | ./dw_sql/cadena_fibra/load_historico_fibra/load_historico_fibra.sql | 40698 |

Tabla 0-3: Código fuente.

Archivos de la marcha atrás

| N. | Nombre (ruta y nombre de archivo) | Checksum |
|----|--|----------|
| 1 | ./instalar/dwh_fijo_crea_campos_cartera_ma.sql | 60728 |
| 2 | ./instalar/dwh_fijo_crea_campos_historico_ma.sql | 59980 |
| 3 | ./instalar/dwh_fijo_crea_carpetas_ma.sh | 24629 |

Tabla 0-4: Archivos marcha atrás.

Instalación y configuración

1. Hacer ftp en binario a la maquina DWH (IP ***) del fichero “dwh_fijo.tar” empleando el usuario usr en la ruta /opt/dwh
2. Descomprimir el fichero “dwh_fijo.tar” con el comando “tar -xvf dwh_fijo.tar” desde el mismo directorio (/opt/ dwh) para desempaquetar los .sql, .caja, .cfg y .sh.
3. Entrar en el directorio /instalar empleando el comando “cd ./instalar” desde el directorio (/opt/dwh).
4. Abrir una sesión de base de datos con el usuario de base de datos usr@DWH, y ejecutar los siguientes ficheros SQL, en el siguiente orden:

@dwh_fijo_create_tmp_carga_solicitud.sql

@dwh_fijo_crea_campos_historico.sql

@dwh_fijo_crea_indice_historico.sql

@dwh_fijo_analiza_historico.sql

@dwh_fijo_crea_campos_cartera.sql

@dwh_fijo_crea_indices_cartera.sql

@dwh_fijo_analiza_cartera.sql

@dwh_fijo_drop_tmp_carga_solicitud.sql

@dwh_fijo_regulariza_bajas_cartera_cdt.sql

5. Cerrar la sesión de base de datos con el comando “exit” y ejecutar el siguiente .sh:
dwh_fijo_crea_carpetas.sh

Borrado de tablas backups

Durante el proceso de instalación se ha tomado la precaución de crear tablas *backups* de Cartera e Historico en caso de ser necesaria una recuperación de datos. Tras comprobar que el pase a producción se ha efectuado correctamente se deberán borrar dichas tablas *backups*. Esto se hará efectivo 4 días tras la realización del pase. A continuación, se detalla el proceso de borrado de las tablas:

1. Entrar en la ruta /opt/ dwh/instalar/ usando el comando “cd /opt/dwh/instalar/”
2. Abrir una sesión de base de datos con el usuario de base de datos usr@dwh los siguientes ficheros sql:

@dwh_fijo_drop_cartera_cdt.sql

@dwh_fijo_drop_historico_cdt.sql

3. Salir de la máquina de Oracle con el comando “exit”.

Descripción de la marcha atrás

1. Si no existe el fichero “dwh_fijo_ma.tar”, hacer ftp en binario a la maquina DWH (IP ***) del fichero “dwh_fijo_ma.tar” empleando el usuario usr. Si existe pasar al segundo paso.
2. Descomprimir el fichero “dwh_fijo_ma.tar” con el comando “tar -xvf dwh_fijo_ma.tar” desde el mismo directorio (/opt/dwh) para desempaquetar los .sql, .caja, .cfg y .sh.
3. Entrar en el directorio ./instalar empleando el comando “cd ./instalar” desde el directorio (/opt/dwh).
4. Ejecutar en la máquina de Oracle dentro de UNIX con el usuario de base de datos dwh los siguientes ficheros SQL, en ese orden:

./instalar/dwh_fijo_crea_campos_historico_ma.sql

./instalar/dwh_fijo_crea_campos_cartera_ma.sql

A través de los siguientes comandos (después de haber entrado en sqlplus):

@dwh_fijo_crea_campos_historico_ma.sql

@dwh_fijo_crea_campos_cartera_ma.sql

5. Salir de la máquina de Oracle con el comando “exit” y una vez fuera, ejecutar el siguiente .sh: ./dwh_fijo_crea_carpetas_ma.sh.

